UNIVERSIDAD CARLOS III DE MADRID

Vision-Based Navigation System for Unmanned Aerial Vehicles

Author:
Abdulla H. Al-Kaff

Supervisor:
Prof. José María Armingol
Prof. Arturo de la Escalera

Intelligent Systems Lab (LSI)
Department of Systems Engineering and Automation

A dissertation submitted in fulfilment of the requirements for the degree of Doctor of Philosophy in

Electric Engineering, Electronics and Automation

Leganés, October, 2017

This dissertation has been approved in partial fulfillment of the degree of: Doctor of Philosophy by the Department of Systems Engineering and Automation - Polytechnic School at the Universidad Carlos III de Madrid on October 5, 2017.

1. Chair: .....
Position: .....
Affiliation: .....
Date and Signature

2. External Examiner: .....
Position: .....
Affiliation: .....
Date and Signature

3. Internal Examiner: .....
Position: .....
Affiliation: .....
Date and Signature

Day of the defense:

Signature from head of PhD committee:

"If the facts do not fit the theory, change the facts."

Albert Einstein

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other University. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text. This dissertation contains less than 60,000 words including appendices, bibliography, footnotes, tables and equations and has less than 100 figures and less than 30 tables.

Abdulla H. Al-Kaff

October, 2017

Acknowledgements

Firstly, I would like to express my sincere gratitude and love to my parents, my sister and my brother; for their endless love, care, support and motivation to improve myself in my life and my work, I cannot say thank you enough for staying with me, just I love you.

Besides, I would like to express my special appreciation and thanks to my supervisors Prof. Arturo de la Escalera and Prof. José María Armingol for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

Last but not least, an honorable mention goes to my partners and friends especially Dr. Fernando García and Dr. David Martín, who supported me in a lot of things, and incented me to strive towards my goal.

Abstract

The main objective of this dissertation is to provide Unmanned Aerial Vehicles (UAVs) with a robust navigation system; in order to allow the UAVs to perform complex tasks autonomously and in real-time. The proposed algorithms deal with solving the navigation problem for outdoor as well as indoor environments, mainly based on visual information that is captured by monocular cameras. In addition, this dissertation presents the advantages of using the visual sensors as the main source of data, or complementing other sensors in providing useful information; in order to improve the accuracy and the robustness of the sensing purposes.

The dissertation mainly covers several research topics based on computer vision techniques: **(I) Pose Estimation**, to provide a solution for estimating the 6D pose of the UAV. This algorithm is based on the combination of SIFT detector and FREAK descriptor; which maintains the performance of the feature points matching and decreases the computational time. Thereafter, the pose estimation problem is solved based on the decomposition of the *world-to-frame* and *frame-to-frame* homographies. **(II) Obstacle Detection and Collision Avoidance**, in which, the UAV is able to sense and detect the frontal obstacles that are situated in its path. The detection algorithm mimics the human behaviors for detecting the approaching obstacles; by analyzing the size changes of the detected feature points, combined with the expansion ratios of the convex hull constructed around the detected feature points from consecutive frames. Then, by comparing the area ratio of the obstacle and the position of the UAV, the method decides if the detected obstacle may cause a collision. Finally, the algorithm extracts the collision-free zones around the obstacle, and combining with the tracked waypoints, the UAV performs the avoidance maneuver. **(III) Navigation Guidance**, which generates the waypoints to determine the flight path based on environment and the situated obstacles. Then provide a strategy to follow the path segments and in an efficient way and perform the flight maneuver smoothly. **(IV) Visual Servoing**, to offer different control solutions

(Fuzzy Logic Control (FLC) and PID), based on the obtained visual information; in order to achieve the flight stability as well as to perform the correct maneuver; to avoid the possible collisions and track the waypoints.

All the proposed algorithms have been verified with real flights in both indoor and outdoor environments, taking into consideration the visual conditions; such as illumination and textures. The obtained results have been validated against other systems; such as VICON motion capture system, DGPS in the case of pose estimate algorithm. In addition, the proposed algorithms have been compared with several previous works in the state of the art, and are results proves the improvement in the accuracy and the robustness of the proposed algorithms.

Finally, this dissertation concludes that the visual sensors have the advantages of lightweight and low consumption and provide reliable information, which is considered as a powerful tool in the navigation systems to increase the autonomy of the UAVs for real-world applications.

Resumen

El objetivo principal de esta tesis es proporcionar Vehículos Aéreos no Tripulados (UAVs) con un sistema de navegación robusto, para permitir a los UAVs realizar tareas complejas de forma autónoma y en tiempo real. Los algoritmos propuestos tratan de resolver problemas de la navegación tanto en ambientes interiores como al aire libre basándose principalmente en la información visual captada por las cámaras monoculares. Además, esta tesis doctoral presenta la ventaja de usar sensores visuales bien como fuente principal de datos o complementando a otros sensores en el suministro de información útil, con el fin de mejorar la precisión y la robustez de los procesos de detección.

La tesis cubre, principalmente, varios temas de investigación basados en técnicas de visión por computador: **(I) Estimación de la Posición y la Orientación (Pose)**, para proporcionar una solución a la estimación de la posición y orientación en 6D del UAV. Este algoritmo se basa en la combinación del detector SIFT y el descriptor FREAK, que mantiene el desempeño del a función de puntos de coincidencia y disminuye el tiempo computacional. De esta manera, se soluciona el problema de la estimación de la posición basándose en la descomposición de las homografías *mundo a imagen* e *imagen a imagen*. **(II) Detección obstáculos y elusión colisiones**, donde el UAV es capaz de percibir y detectar los obstáculos frontales que se encuentran en su camino. El algoritmo de detección imita comportamientos humanos para detectar los obstáculos que se acercan, mediante el análisis de la magnitud del cambio de los puntos característicos detectados de referencia, combinado con los ratios de expansión de los contornos convexos construidos alrededor de los puntos característicos detectados en frames consecutivos. A continuación, comparando la proporción del área del obstáculo y la posición del UAV, el método decide si el obstáculo detectado puede provocar una colisión. Por último, el algoritmo extrae las zonas libres de colisión alrededor del obstáculo y combinándolo con los puntos de referencia, el UAV realiza la maniobra de evasión. **(III) Guiado de**

navegación, que genera los puntos de referencia para determinar la trayectoria de vuelo basada en el entorno y en los obstáculos detectados que encuentra. Proporciona una estrategia para seguir los segmentos del trazado de una manera eficiente y realizar la maniobra de vuelo con suavidad. **(IV) Guiado por Visión**, para ofrecer soluciones de control diferentes (Control de Lógica Fuzzy (FLC) y PID), basados en la información visual obtenida con el fin de lograr la estabilidad de vuelo, así como realizar la maniobra correcta para evitar posibles colisiones y seguir los puntos de referencia.

Todos los algoritmos propuestos han sido verificados con vuelos reales en ambientes exteriores e interiores, tomando en consideración condiciones visuales como la iluminación y las texturas. Los resultados obtenidos han sido validados con otros sistemas: como el sistema de captura de movimiento VICON y DGPS en el caso del algoritmo de estimación de la posición y orientación. Además, los algoritmos propuestos han sido comparados con trabajos anteriores recogidos en el estado del arte con resultados que demuestran una mejora de la precisión y la robustez de los algoritmos propuestos.

Esta tesis doctoral concluye que los sensores visuales tienen las ventajes de tener un peso ligero y un bajo consumo y, proporcionar información fiable, lo cual lo hace una poderosa herramienta en los sistemas de navegación para aumentar la autonomía de los UAVs en aplicaciones del mundo real.

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

AAR Autonomous Aerial Refueling. 20–22

ACS Ant Colony System. 20

AUV Autonomous Underwater Vehicle. 25, 27

BRIEF Binary Robust Independent Elementary Features. 31, 43, 44

BRISK Binary Robust Invariant Scalable Keypoints. 43

BRR Boom-and-Receptacle Refueling. 20, 22

CMOS Complementary Metal-Oxide-Semiconductor. 81

DCT Discrete Cosine Transform. 34

DGPS Differential Global Positioning System. x, xii, 13, 25, 26, 49, 61, 73–75, 80

DOF Degrees Of Freedom. 13, 39

DOG Difference of Gaussian. 44

EKF Extended Kalman Filter. 26, 27, 29, 32, 33, 35

FAST Features from Accelerated Segment Test. 31, 43, 44

FLC Fuzzy Logic Control. x, xii, 12, 101, 104–106, 132

FOV Field Of View. 19, 85, 86, 98, 99

FREAK Fast Retina Keypoint. ix, xi, 10, 13, 43–45, 50–54, 56–62, 79, 80, 134

GA Genetic Algorithm. 20

GCS Ground Control Station. 1

GNSS Global Navigation Satellite System. 23

GPS Global Positioning System. 7–9, 12, 18, 19, 21, 25–27, 32, 73, 136

HMT Hidden Markov Trees. 28

HSV Hue, Saturation, and Value. 21, 22

IBVS Image-Based Visual Servoing. 23

IDCT Inverse Discrete Cosine Transform. 34

IFF Identification Friend-or-Foe. 27

IMU Inertial Measurement Unit. 12, 18, 25–27, 30–33, 35, 73, 106, 110, 112, 136

INS Inertial Navigation System. 8, 21, 26, 27

IR Infrared. 9, 18

KLT Kanade-Lucas-Thomasi. 37

LIDAR Light Detection And Ranging. 18

LK Lucas-Kanade. 20, 31

LMedS Least Median Square Estimator. 20

LOG Laplacian of Gaussian. 45

LQG Linear Quadratic Gaussian. 28, 36

LQS Linear Quadratic Servo. 23

LQT Linear Quadratic Tracking. 36

LTR Loop Transfer Recovery. 28

MAV Micro Unmanned Aerial Vehicle. 6, 7, 9, 27, 31, 34, 81

MEMS Micro-Electro-Mechanical System. 22

MER Mars Exploration Program. 30

MonoSLAM Mono-Simultaneous Localization And Mapping. 28

MOPS Multi-scale Oriented-Patches. 34

MRF Markov Random Field. 35

MTOW Mean Take off Weight. 1, 5, 6

OOI Object of Interest. 89

ORB Oriented FAST and Rotated BRIEF. 19, 43

PDR Probe-and-Drogue Refueling. 21

PID Proportional-Integral-Derivative. 18, 35–37, 102

PPBVS Partial Posed Based Visual Servoing. 23

PTAM Parallel Tracking and Mapping. 28, 29

RANSAC RANdom SAmple Consensus. 30, 63

ROI Region Of Interest. 85, 98

RPAS Remotely Piloted Aerial System. 1

RPM Revolutions per minute. 147, 149

RTK Real Time Kinematic. 18

SA Simulated Annealing. 102–104, 135

SIFT Scale-Invariant Feature Transform. ix, xi, 10, 13, 31, 32, 34, 37, 43–45, 50–62, 79, 80, 86, 134

SINS Strap-down Inertial Navigation System. 26

SLAM Simultaneous Localization And Mapping. 19, 22, 25, 28–30

SNS Satellite-based Navigation System. 25

SoG Sum of Gaussian. 29

SUAV Small Unmanned Aerial Vehicle. 6, 7, 81

SURF Speeded Up Robust Features. 13, 20, 43–46, 50–54, 56–61

SVD Singular Value Decomposition. 66, 69

SWaP Size, Weight and Power. 7, 81

UAV Unmanned Aerial Vehicle. ix–xii, 1–10, 12, 13, 15–29, 31–40, 49, 61, 62, 72–74, 77, 80, 82–85, 90, 93–99, 101, 102, 106, 107, 109, 110, 112, 115, 116, 119–130, 132–137, 143

UGV Unmanned Ground Vehicle. 18–20, 25, 27

UKF Unscented Kalman Filter. 26

VO Visual Odometry. 28, 30, 31, 40

VS Visual Servoing. 36

VSLAM Visual Simultaneous Localization And Mapping. 28–30, 40

VTOL Vertical Take-Off and Landing. 6, 18

CHAPTER 1

Introduction

Unmanned Aerial Vehicle (UAV), Remotely Piloted Aerial System (RPAS) or what is commonly known as a drone is the term that describes the aircraft platform without a human pilot on-board. UAV can be either teleoperated remotely by the pilot in the Ground Control Station (GCS) or autonomously using the on-board sensors mounted on it, following preprogrammed operations. However, this terminology not only refers to the vehicle itself, but also to all of the supporting hardware and software; including sensors, micro-controllers, ground stations, communication protocols and user interfaces [38].

There are many classification schemes that have been presented in order to categorize the UAVs, however, there is no unique scheme of classification. These schemes are based on a large number of different characteristics; such as drone mass, size, mission range, operation altitude, operation duration, Mean Take off Weight (MTOW), flying principle, propulsion mode, operation condition, drone capability and combination of these characteristics.

Figure 1.1 shows the three main categories and models of UAVs based on its body shape and flying principles. In addition, the advantages and disadvantages and the purposes of each category of the UAVs are summarized in Table 1.1.

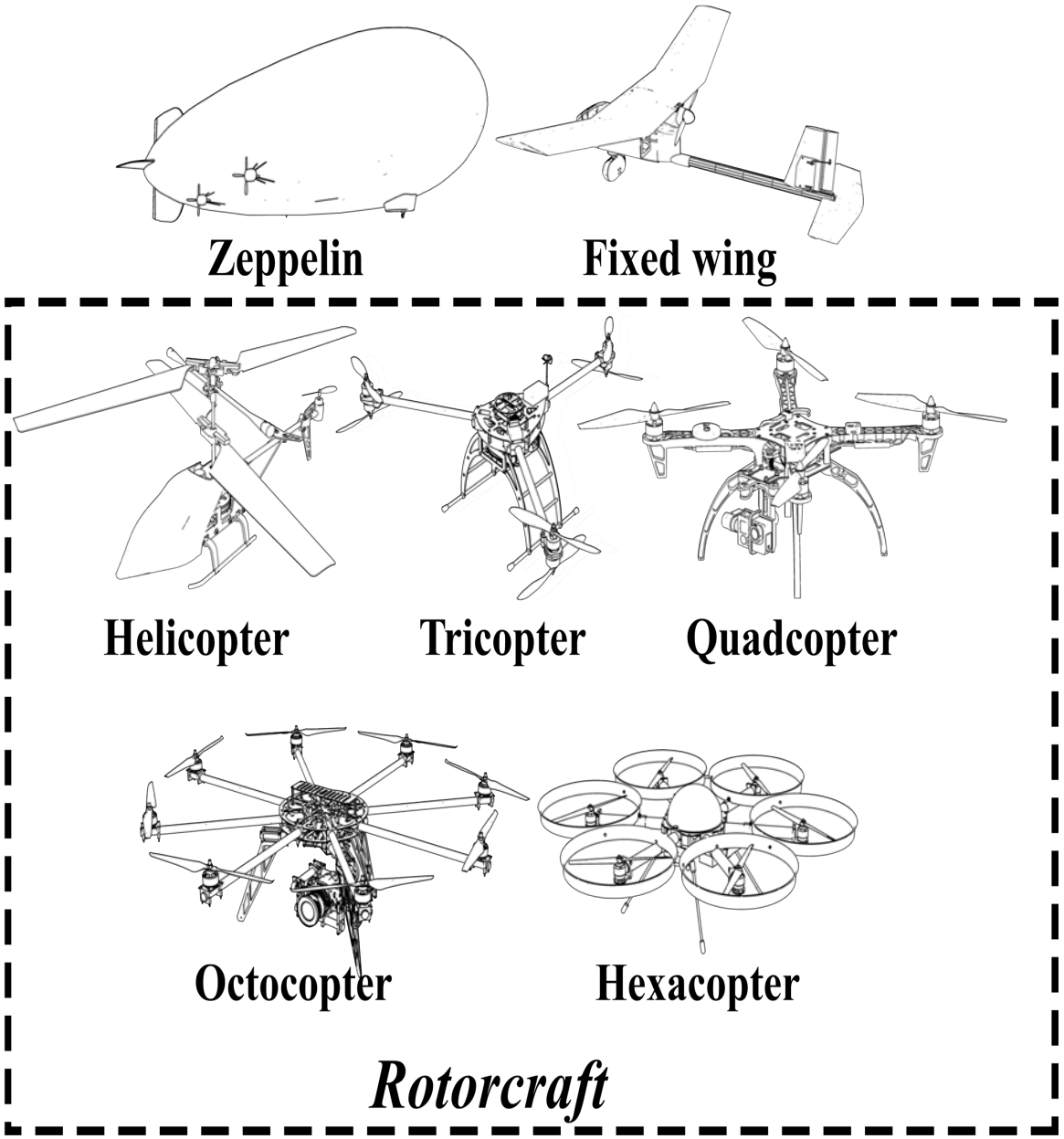


Figure 1.1 UAV Models.

One of the detailed and widely used schemes has been proposed by [302], as shown in Table 1.2. In which, the UAVs are classified based on the mass, range, altitude, and endurance.

Table 1.1 Advantages and Disadvantages of the UAVs Categories

Category	Advantages	Disadvantages	Purposes	Price (€)
Fixed-Wing	• Long endurance • Large area coverage • Fast flight speed	• Require a space for Take-off and Landing • Disability of VTOL and hover • Require more training • Expensive	• Aerial Mapping • Pipeline and Power line inspection • Agricultural purposes • Surveillance • Racing	50-120000
Fixed-Wing Hybrid	– VTOL – long-endurance flight	– In development	– Delivery – Aerial Mapping	TDB
Single-Rotor	• VTOL and hover flight • Long endurance • More payload capability	• More dangerous • Require more training • Expensive	• Aerial Photography • Surveillance	50-120000
Multi-Rotor	– Accessibility – Ease of use – VTOL and hover flight – High flight stability – Small area maneuvers	– Short endurance – Limited payload capacity	– Aerial Photography – Infrastructure inspection – Surveillance – Racing	50-65000

Table 1.2 Classification of UAVs based on Mass, Range, Altitude and Endurance (SOURCE [302])

Category	Mass (kg)	Range (km)	Flight altitude (m)	Endurance (h)
Micro	<5	<10	250	1
Mini	<20/25/30/150[a]	<10	150/250/300	
Tactical				
Close range (CR)	25–150	10–30	3000	2–4
Short Range (SR)	50–250	30–70	3000	3–6
Medium Range (MR)	150–500	70–200	5000	6–10
MR endurance (MRE)	500–1500	>500	8000	10–18
Low altitude deep penetration (LADP)	250–2500	>250	50–9000	0.5–1
Low altitude long endurance (LALE)	15–25	>500	3000	>24
Medium altitude long endurance (MALE)	1000–1500	>500	3000	24–48
Strategic				
High altitude long endurance (HALE)	2500–5000	>2000	20000	24–48
Stratospheric (Strato)	>2500	>2000	>20000	>48
Exo-stratospheric (EXO)	TBD	TBD	>30500	TBD
Special task				
Unmanned Combat AV (UCAV)	>1000	1500	12000	2
Lethal (LET)	TBD	300	4000	3–4
Decoys (DEC)	150–250	0–500	50–5000	<4

[a] Varies with national legal restrictions.

Moreover, another scheme based on MTOW and the ground impact risk has been proposed by [81], as shown in Table 1.3.

Although UAVs were designed and supported originally for the defense and military purposes; such as aerial attacks or military air cover, to avoid the risk of human lives. During the last decade, with the developments in microelectronics and

the increase in the computing efficiency, Small Unmanned Aerial Vehicles (SUAVs) and Micro Unmanned Aerial Vehicles (MAVs) have encountered a significant focus among the robotics research community.

Table 1.3 Classification of UAVs based on the MTOW and the ground impact risk (SOURCE [81])

Category Number	T_{GI} [a]	MTOW	Name	Note
0	10^2	Less than 1 kg	Micro	• Most countries do not regulate this category since these vehicles pose minimal threat to human life or property.
1	10^3	Up to 1 kg	Mini	– These two categories roughly correspond to R/C model aircraft
2	10^4	Up to 13.5 kg	Small	
3	10^5	Up to 242 kg	Light/ ultralight	• Airworthiness certification for this category may be based either on ultralights (FAR [b]Part 103), LSA [c](Order 8130), or even normal aircraft (FAR Part 23)
4	10^6	Up to 4332 kg	Normal	– Based on MTOW these vehicles correspond to normal aircraft (FAR Part 23)
5	10^7	Over 4332 kg	Large	• These vehicles best correspond to the transport category (FAR Part 25)

[a] T_{GI} is the minimum time between ground impact accidents.
[b] Federal Aviation Regulations.
[c] Light Sport Aircraft.

Moreover, due to their ability to operate in remote, dangerous and dull situations, especially helicopters and Vertical Take-Off and Landing (VTOL) rotor-craft systems; for example quad/hexa/octo-rotors, they are increasingly used in several civil and scientific applications; such as surveying and mapping, rescue operation

in disasters [4, 97], spatial information acquisition, buildings inspection [16, 68, 99], data collection from inaccessible areas, geophysics exploration [102, 150], traffic monitoring [151], animal protection[1] [324], agricultural crops monitoring [25], manipulation and transportation[2] [222] or navigation purposes [43, 78, 303].

Nowadays, with the current technology and the variety and the complexity of the tasks, modern UAVs aim at higher levels of autonomy and performing flight stabilization. The main part of an autonomous UAV is the navigation system and its supporting subsystems. The autonomous navigation system utilizes information from various subsystems in order to achieve three essential tasks: to estimate the pose of the UAV (position and orientation) **(Localization)**, to identify obstacles in the surrounding and act in consequence in order to avoid them **(Obstacle detection and avoidance)**, and to send commands to stabilize the attitude and follow guidance objectives **(Control loop)**.

The difficulty appears due to working with SUAVs or MAVs; such as Ar.Drone Parrot [3], DJI Pantom series [4], AscTec Hummingbird [5], Voyager 3 [6], 3DR SOLO [7] or TALI H500 [8], that is due to the Size, Weight and Power (SWaP) constraints That is because of the size of these vehicles is getting smaller (few centimeters) and the weight is getting lighter (few grams), which leads to a significant limitation in the payload capabilities and the power consumption. Therefore, with these properties, mounting on-board sensors that are helpful for the navigation purposes is considered a challenging problem.

In outdoor operations, most of the navigation systems are based on Global Positioning System (GPS) [139, 160, 294] to locate their position. In these systems, the precision depends directly on the number of the satellites connected. However, the Global Positioning System (GPS)-based systems do not provide reliable solutions in GPS-denied environments; such as urban areas, forests, canyons or low altitude flights that can reduce the satellite visibility. Furthermore, in other scenarios, like indoor operations, GPS loses totally its efficiency because of the absence of the information.

[1] http://www.bbc.com/news/business-28132521
[2] http://www.service-drone.com/en/production/logistics-and-transport
[3] http://ardrone2.parrot.com/
[4] http://www.dji.com/products/phantom
[5] http://www.asctec.de/en/uav-uas-drones-rpas-roav/asctec-hummingbird/
[6] http://www.walkera.com/en/products/aerialdrones/voyager3/
[7] https://3dr.com/solo-drone/
[8] http://www.walkera.com/en/products/aerialdrones/talih500/

UAVs need a robust positioning system to avoid catastrophic control actions, which can be caused by errors in the position estimation, so that, different approaches are proposed to solve this problem. Using the information provided by the GPS combined with the data obtained by the Inertial Navigation System (INS) is one of the most popular approaches, at which, the data of the INS and the GPS are fused together to minimize the position error [23, 37, 235, 334]. However, two main drawbacks appeared of this approach: First, the dependency on external satellite signals, and second, the lack of precision of the estimated position, which is based on the INS data.

Therefore, some specific solutions have been provided such as using radar [253] or laser sensor [51, 123]. However, these sensors require more payload capabilities and higher power consumption. On the other hand, other approaches use some expensive external localization systems such as VICON systems [51, 218, 222] in order to capture the motion of the UAV in the case of indoor environments.

Owing to its capability to provide detailed information about the surrounding environments, computer vision algorithms and visual sensors play a vital role as the main solution in indoor and outdoor scenarios [45, 150, 169, 230]. In addition, visual sensors can be used as stand-alone sensors or combined with other sensors to improve the accuracy and robustness of the navigation system.

Visual sensors, such as cameras, have the advantage of lightweight, low power consumption and relatively low-cost. In addition, visual sensors, provide rich information of the environment, which can be processed and applied to real-time applications. However, the accuracy of these approaches depends on different factors; such as images resolution, capturing time, viewing angle, illumination, different structures of aerial images and reference data.

From these aspects, this dissertation presents visual navigation system based on the on-board monocular cameras. At which, pose estimation, obstacle detection, avoidance decision, and control algorithms are proposed. This navigation system provides robust solution for UAVs in real-time for both outdoor and indoor scenarios.

1.1 Problem Statement

Because of the behavior of the missions and the types of the tasks and the applications of the UAVs and the complexity of the maneuvers required, there is a neces-

sity of increasing the capabilities of the navigational systems of the UAVs with high levels of robustness and accuracy.

Autonomous navigational systems in unknown and GPS-denied environments are still an open research problem. Moreover, this problem is getting more challenging with the MAVs due to the limitation in the power and payload constraints. Therefore, using visual sensors such are cameras for the navigation purposes are often preferable.

The problem of understanding the world in its three dimensions using visual sensors is very challenging and critical. Different researchers tend to work with stereo cameras; such as Bumblebee series [9], ZED [10] or DUO series [11] in order to estimate the three dimensions information of the surrounding environment. However, using stereo cameras on MAVS constrains the length of the baseline, which in turn reduces the accuracy of depth estimation.

Another proposed solution is working with RGB-D sensors; such as Microsoft Kinect 360 [12], Kinect *V2* [13], Asus Xtion [14], or Structure Sensor [15] in order to deal with the problem of the depth estimation [29, 30, 156, 174, 267, 277, 301]. These sensors work well in indoor environments. However, the problem appears of working in outdoor environments, because this type of sensors is based on the projection and analyzing a speckle pattern of the Infrared (IR), which faults and provides unreliable information in outdoor environments.

Moreover, the relatively high cost (stereo cameras), heavy weight and large size of these sensors do not make it practical to be used with MAVs.

On the other hand, due to its light weight and low power consumption and the rich information that obtain, monocular cameras are considered as a powerful sensor used in both indoor and outdoor environments.

From this point of view, this dissertation deals with solving the navigation problem of UAVs for indoor and outdoor environments based mainly on monocular vision algorithms, by designing and implementing a robust and efficient positioning system for indoor and outdoor environments with the capabilities of avoiding any unexpected actions. The proposed vision-based navigation system is discussed in

[9] https://www.ptgrey.com/stereo-vision-cameras-systems
[10] https://www.stereolabs.com/
[11] https://duo3d.com/products
[12] http://www.xbox.com/en-US/xbox-360/accessories/kinect
[13] http://www.xbox.com/en-US/xbox-one/accessories/kinect-for-xbox-one
[14] http://www.asus.com/3D-Sensor/Xtion_PRO_LIVE/
[15] http://structure.io/

four topics:

- **Monocular Vision-based Pose Estimation:**
  - Provides a solution to localize and estimate the UAV position and orientation in the three dimensional world.
- **Monocular Vision-based Obstacle Detection and Avoidance**
  - Provides a solution for detecting the approaching obstacles and estimates the position of the obstacle in front of the UAV to take the best decision and avoid the collision with it.
- **Navigation Guidance**
  - Provides a solution for generating and tracking the waypoints, in order to deal with the path planning problem, for the autonomous flights purposes.
- **Visual Servoing**
  - Provides a robust controller based on the visual information as inputs in order to maintain the stability of the UAV, and sends commands for certain maneuvers.

1.2 Dissertation Proposals

The main objective of this dissertation is to propose a robust navigation system for UAVs, which is feasible and efficient for indoor and outdoor environments. As shown in Figure 1.2, the whole system is divided into two main parts: **Vision-based Navigation Guidance** to solve the localization and obstacle detection; and **Motion Control** to deal with the flight control problem. The proposed algorithms are implemented for the following objectives:

- Estimates the position (x, y, z) and orientation (ϕ, θ, ψ) of the UAV by solving the different homographies *world-to-frame* and *frame-to-frame*, using the 2D images captured by a single downward-looking camera. This system is based on the SIFT detector and FREAK descriptor combination, which can increases the performance of feature matching and tracking, and decrease the computational time.

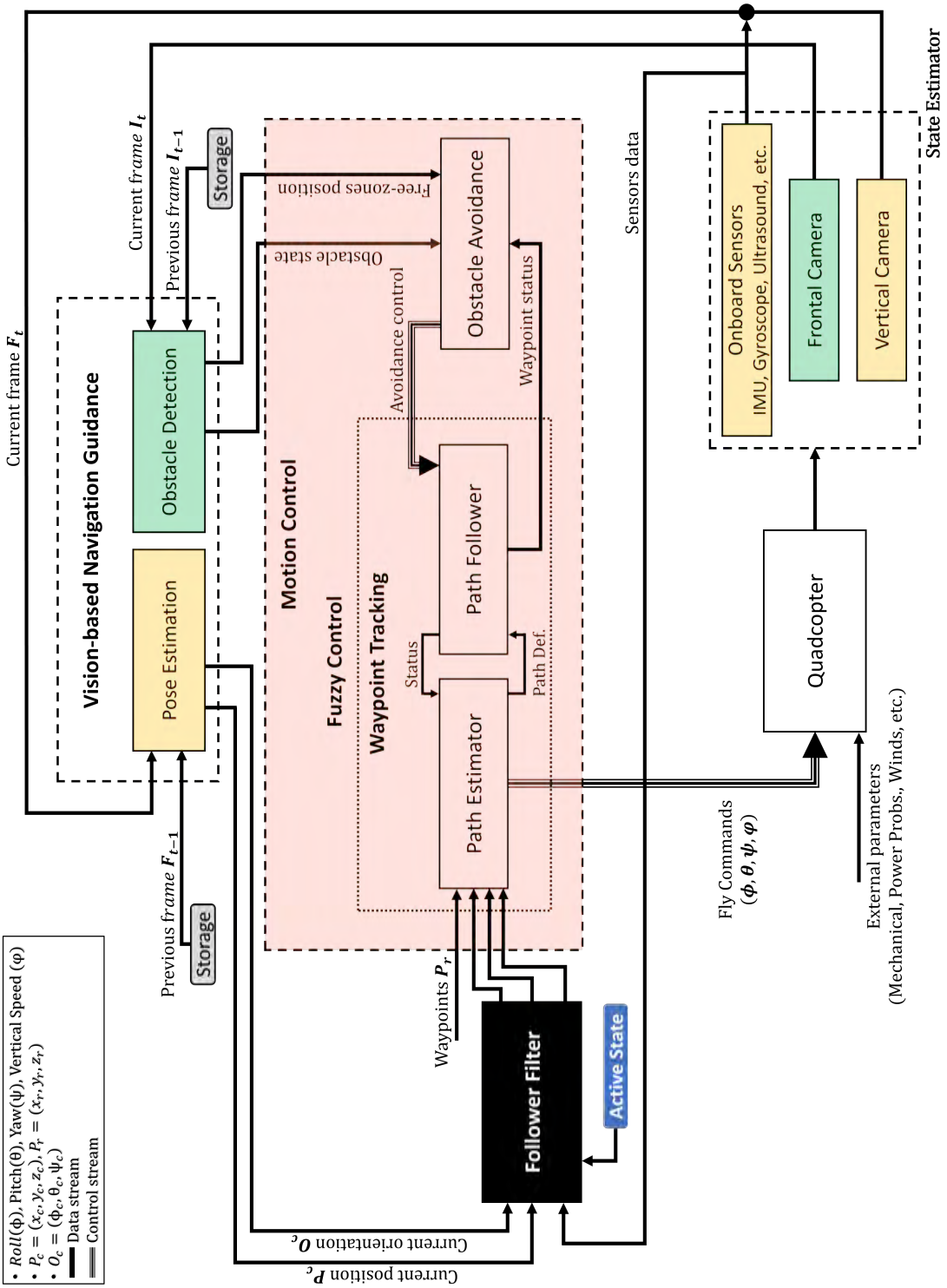


Figure 1.2 General overview of the Proposed System.

- Proposes a bio-inspired method that mimics the human behavior of analyzing the size states of the approaching obstacles using monocular camera. This method is divided into two main stages: **Vision-based Navigation Guidance**, where the obstacle detection algorithm is performed based on the obtained information from the images captured from the frontal camera; and **Motion Control**, where the avoidance decision is estimated and send the control commands to the UAV.

- Presents an approach in order to perform autonomous flights from the start point to the destination. The presented approach is divided into two stages; **Path Planning:** a robust guidance algorithm, that generates the waypoints to determine the flight path based on environment, and the situated obstacles. **Waypoint Tracking:** a strategy of following and swapping the generated waypoints efficiently.

- Implements Fuzzy Logic Control (FLC) based mainly on the vision information and combined with data from other on-board sensors; such as Inertial Measurement Unit (IMU), GPS and ultrasonic; in order to deal with the mission of waypoint tracking, obstacle avoidance actions, and the flight stability.

1.3 Dissertation Contributions and Outline

This dissertation demonstrates the efficiency and the performance of the computer vision algorithms that are used in UAVs. In the dissertation, different vision-based algorithms are proposed to solve the problems of pose estimation, obstacle detection, collision avoidance and flight control. This is, in order to improve the navigation systems for autonomous UAVs in indoor and outdoor environments.

The contributions are detailed within the organization of this dissertation as follows:

- **Chapter 2: State-of-the-Art**

  - A literature review of the computer vision-based navigation systems algorithms and techniques is presented in this chapter. This detailed review will focus on the algorithms related to the Pose Estimation, Obstacle Detection, Collision Avoidance and Visual Control applied on UAVs.

- **Chapter 3: Monocular Vision-based Pose Estimation**
  - This chapter presents the feature-based pose estimation algorithm, and divided into main parts:
    - Present a performance evaluation of the two robust feature detection and description algorithms (SIFT and SURF), with the effect of combining the FREAK descriptor. In addition, the results show the best choice of detector and descriptor to be used in real-time navigation systems with high level of accuracy.
    - Homographies formulation (*frame-to-frame* and *world-to-frame*), translation and rotation matrices extraction, and the six Degrees Of Freedoms (DOFs) pose estimation are explained.
  - Results of the proposed algorithm are presented with a comparison to DGPS in outdoor and VICON system in indoor environments.
- **Chapter 4: Monocular Vision-based Obstacle Detection**
  - Presents the bio-inspired algorithm for the detection process. Where, the presence and the position of the approaching frontal obstacles are estimated, based on the analysis of the size expansion ratios from a sequence of two-dimensional frames.
  - Describes the combined mission of the waypoint tracking and the avoidance algorithm. Based on the UAV position, next waypoint, and the position of the detected obstacle, the UAV is able to take the action accurately to avoid the possible collision.
- **Chapter 5: Visual Servoing for UAVs**
  - Introduces the flight stability control based on fuzzy logic controller. This controller improves the stability of the UAV in hovering mode, and the maneuver control for waypoint tracking based mainly on the data estimated from the camera, and the data obtained by the other sensors.
  - Presents a collision avoidance algorithm based on PID control, in order to change the direction of the UAV once a frontal obstacle is detected, thereafter recalculate the new position to return to the original path.

- **Chapter 6: Conclusions and Future Work**
  - Summarizes the final conclusion of the proposed algorithms for the navigation system, and discusses the possibilities of the future works.

CHAPTER 2

State-of-the-Art

Computer vision plays a vital role in the most of the Unmanned Aerial Vehicles applications[1]. These applications vary from a simple aerial photography, to very complex tasks; such as rescue operations, aerial refueling or inspections. All of these tasks require high level of accuracy in order to provide reliable decision and maneuver tasks.

Aerial imagery or aerial filming is considered as one of the basic and demanding applications; such as filming sports games [2], events [3] or even weddings [4].

[1] Publications of the author related to the chapter:

- Survey of Computer Vision Algorithms and Applications for Unmanned Aerial Vehicles, Journal of Expert Systems with Applications [14].

[2] The Future of Sports Photography: Drones
[3] Airborne camera makes concert scene
[4] Camera in the Sky: Using Drones in Wedding Photography and Videos

Recently, with the advances in computer vision algorithms, technologies and sensors, the concept of using aerial images just for photography and filming was changed to be used widely in more complex applications; such as thematic and topographic mapping of the terrain [8, 77, 184, 199, 292]; exploration of un reachable areas such as islands [332], rivers [254], forests [78, 338] or oceans [289, 290]; surveillance purposes [112, 119, 186, 274]; and search and rescue operations after catastrophes [97, 171, 307].

Another widely demanded application that takes the advantages of the aerial imagery over the traditional sensing, is the traffic monitoring [151, 252]. Traffic monitoring using UAVs includes the estimation of the traffic flow behavior [130, 162], traffic speed [153], roads state [191, 351], in addition to the emergencies and car accidents [250].

Furthermore, wide variety of different autonomous applications have been presented; such as autonomous take-off and landing [54, 113, 147, 178, 265], autonomous aerial refueling [1, 56, 321, 325, 331], autonomous tracking [3, 190, 211, 350] or autonomous route planning [119, 168, 266, 327, 328], where high levels of accuracy of localization, detection and tracking are required.

Several surveys that cover different computer vision concepts, techniques and applications that are related to UAVs are presented in [57] (visual servoing), [193] (aerial surveillance and multi-UAV cooperation), [4] (disaster research), [151] (traffic monitoring), and [339] (forest fire monitoring).

This chapter presents a literature review of the UAV applications, the algorithms and the techniques that are mainly based on the computer vision applied on the UAVs. In addition, demonstrates the efficiency of the visual devices as a main or complementary sensor, that provides information about the environment for the purposes of the UAVs navigation systems.

With this increase in the levels of autonomy, flight stabilization and the complexity of the tasks, robust and efferent navigation system is required. This dissertation focuses on the navigation systems that are based on the visual information obtained by monocular cameras. The navigation systems is divided into three main subsystems: **Pose estimation** which aims to estimate the position and the attitude of the UAV in two and three dimensional representations, **Obstacle detection and avoidance** that detects and feeds back the position of the obstacles that are situated in the path of the UAV, and finally the **Visual servoing** subsystem, at which, the maneuver commands are managed and sent in order to maintain the flight stability

and following the path.

The following sections discuss the evaluation of vision-based algorithms, methods and techniques that are related to the UAVs navigation systems in the last decade. In addition, it presents the most modern and demanded applications that are based on computer vision.

2.1 Vision-based Autonomous Applications for UAVs

The fields of computer vision and image processing have shown a powerful tool in different applications for UAVs. Autonomous UAVs applications are an interesting area, but at the same time are considered as a challenging subject. Among these applications, this literature throws light on the autonomous applications for take-off and landing, surveillance and security, aerial refueling, and inspection as shown in Table 2.1.

2.1.1 Autonomous Landing

Autonomous Take-off and Landing is a fundamental task not only for VTOL vehicles [72, 110] but also for fixed wings UAVs [137, 154]. For vision-based take-off and landing, different solutions have been proposed in order to deal with this problem [75, 131, 178, 323].

In [39], an implementation method, based on backstepping controller for autonomous take-off and landing, on stationary landing-pad was presented. In this method, a Kalman filter was used to fuse the data from the IMU and GPS to estimate the attitude and the speed of the quadcopter. In addition, a Light Detection And Ranging (LIDAR) sensor was used instead of the spatial device, to measure the distance of the landing-pad to the quadcopter. The results show the capability of the quadcopter to perform autonomous take-off and landing. However, the system suffers from the lack of accuracy in determining the attitude of the quadcopter, this is due to the error generated in the IMU and GPS measurements.

Wenzel *et al.* introduced a solution using Wii IR camera [317]. The concept of this approach is to detect four lights LEDs pattern situated on a mobile robot. However, the system is able to track the landing pattern, but the use of IR camera has several limitations; such as that the system cannot be applicable in outdoor flights because of the sensor sensibility to the sunlight. Furthermore, the system has maximum

Table 2.1 Autonomous UAVs applications based on computer vision

Application	Description	Purpose	Related Work
Autonomous Landing	Take-off and Landing	VTOL	[39, 59, 75, 106, 131, 137, 147, 178, 317, 329]
		Fixed Wing	[80, 159, 166, 167, 231, 247, 249]
Autonomous Surveillance	Using aerial imaginary for monitoring and vigilance purposes	Traffic	[79, 130, 153, 162, 252]
		Agricultural crop	[25, 236, 299]
		Animal protection	[312, 324]
		Other	[129, 274, 336]
Aerial Refueling	Refueling the aircrafts during the flight by using a tanker aircraft	Boom-and-Receptacle	[64, 88, 208, 318, 330, 340, 341]
		Probe-and-Drogue	[32, 213, 214, 260, 288, 322, 325]
Inspection	Inspecting the damages and collapses in the surfaces of the infrastructures for monitoring and maintenance purposes	Buildings	[16, 52, 69, 99, 238, 246]
		Bridges	[62, 126, 221]
		Wind turbines	[132, 285, 286]
		Power lines	[26, 40, 58, 87]
Search and Rescue	Gather information in disaster and hazardous sites		[5, 83, 97, 233, 293, 320]
Mapping	Collecting topographical, thematic and geospatial data		[8, 77, 102, 118, 125, 184, 199, 236, 248, 292, 332]

detection region up to 2.5m because of the limitation of the IR cameras.

Another vision-based cooperation between a UAV and an Unmanned Ground Vehicle (UGV) has been presented in [138]. At which, RGB camera is used to detect the landmark instead of using IR sensor like in [317]. This approach used Hough Transform to detect 20cm radius circular landmark attached to the mobile robot. Then the detected circle is restricted by a square shape in order to estimate the center. Finally, a closed-loop Proportional-Integral-Derivative (PID) is applied to perform the control of the UAV.

Working with fixed-wing UAVs, Muskardin *et al.* proposed a cooperative control approach of autonomous landing on a mobile ground vehicle [231]. The concept of this approach aimed to eliminate the use of the landing gear required for

the fixed-wing UAV. In this system, an Real Time Kinematic (RTK)-GPS and IMU sensors are used to estimate the pose and the speed of both vehicles, the UAV equipped by a downward-looking camera to detect and track the landing-pad, and the ground vehicle is equipped by optical markers. In addition, a feedback controller is applied in order to coordinate the motion of both vehicles, and to send velocity commands to both controllers.

In [106], an image-based visual control, coupled with a GPS navigation system have been presented to deal with the problem of autonomous landing on moving UGV. In this system, the attitude of the UAV is estimated using the GPS data, and the distance between the landing-pad and the UAV is determined based on the processed images. The landing-pad is made up of two parts; a $0.6 \times 0.6m$ pattern to be detected from high altitudes, and a $0.2 \times 0.2m$ QR code to guide the UAV when it flies near to the landing-pad. The results show that the system solved the problem of detecting the landing-pad in high or low altitudes, however, the system used the internal controller of the micro-controller. Furthermore, all the experiments were limited to simulations.

Multi-scale Oriented FAST and Rotated BRIEF (ORB) method [259] integrated with the SLAM map to detect the landing site has been presented in [329]. Although the experiments have shown good results, this method is dependent on the map generated from the SLAM and consequently loses its accuracy in the case of the absence of the map.

2.1.2 Autonomous Surveillance

Surveillance based on aerial imaginary is one of the main applications that takes the advantages of UAVs in both military and civil areas. Surveillance can be in battlefields, coasts, borders, forest, highways, urban, and indoor environments. Different methods and approaches have been presented to optimize the solution of the surveillance in terms of time, number of UAVs, autonomy, etc.

Fred *et al.* [103] presented an evaluation approach, comparing the performance of the methods and algorithms, that employed to the UAVs for autonomous surveillance tasks with the guidance of human operators. Recently, a study based on the nature of the tasks and the capabilities of the UAVs has been presented in [79]. In this evaluation study, a scheme of comparing different small UAVs has been proposed; in order to select the adequate UAV that provides the high performance and safety to improve the traffic conformance intelligence.

In [255], an ultra wide angle surveillance system with multi-UAV has been presented. The system is based on stitching method in order to create a panoramic image, if a motion captured in the FOV of the images from multiple UAVs. In addition, a vision feedback formation control framework was presented to adjust the positions of the UAVs during the surveillance process. However, the system is not based on the on-board sensors, but it needs an external motion capture system to estimate the UAV position, which is not applicable for outdoor environments.

A feature-based approach for detecting and tracking multiple moving targets from UAVs was presented in [282]. First, the features are extracted using Harris detector, then the pyramidal Lucas-Kanade (LK) optical flow model, and the Least Median Square Estimator (LMedS) are used in order to classify the movement of the detected features. Finally, a Kalman filter and a template matching algorithm are used to track the detected targets.

UAV-UGV cooperation approach for autonomous indoor surveillance tasks has been presented in [268]. In this system, both vehicles are based on visual information for navigation, localization and landing (UAV). In addition to the helipad that carries the UAV, the UGV is equipped with the sensors necessary for the surveillance tasks. Based on SURF detector, the UGV can detect and track the landmark features from the input images and estimate its location, then move autonomously along predefined waypoints. Once the UGV reaches to an inaccessible location, the UAV flies from the UGV and starts the aerial inspection task. Finally, by using color detection algorithms, the UAV locates the helipad pattern and performs the autonomous landing.

Dealing with the mission planning problem for multiple UAVs, Geng *et al.* [111] proposed an approach that provides continuous surveillance operations. This approach is divided into two phases. The first phase addresses the search of the locations of the cameras in order to provide the complete coverage of the targets in the area. To achieve this, a Genetic Algorithm (GA) is implemented to obtain the optimal solution. The second phase deals with distributing the selected locations that are obtained from GA over a number of UAVs, and creating the paths to be followed in the surveillance. Ant Colony System (ACS) algorithm is used to find the solution for the paths and endurance. However, the experiments have been limited to simulations.

2.1.3 Autonomous Aerial Refueling

Autonomous Aerial Refueling (AAR) describes the process of air-to-air refueling, or in other words, is the in-flight refueling. AAR is divided into two main techniques [181], the first one is Boom-and-Receptacle Refueling (BRR), in which, a single flying tube (boom) is moving from the tanker aircraft for connecting the receptacle, that is situated in the receiver aircraft. The second technique is the Probe-and-Drogue Refueling (PDR), in which, the receiver releases a flexible hose (drogue), and the tanker maintains its position to insert the rigid probe into this drogue. Figure 2.1 shows the concept of the two types of the AAR system. AAR is very critical operation, and usually the tanker pilot has to be well trained to perform these complex operations. On the other hand, in UAVs, the remote controlling for AAR operation increases the complexity of the task. Different techniques use GPS and INS to obtain the relative pose of the tanker with respect to the receiver aircraft. However, these techniques have two drawbacks: First, in certain cases, the GPS data cannot be obtained, especially when the receiver aircraft is bigger than the tanker, and prevents the connection with the satellites. The second drawback is the integration drift of the INS measurements.

On the other hand, the vision-based methods proposed an alternative or complementary solution for AAR. Different studies and surveys of vision-based methods and approaches for AAR, that are used with UAVs have been introduced in [1, 181, 208].

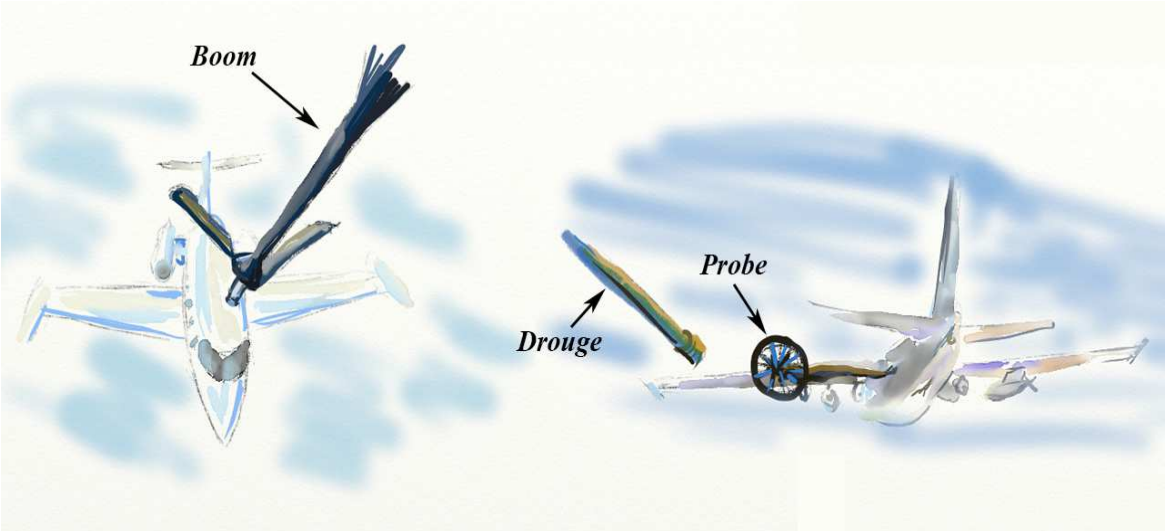


Figure 2.1 Aerial Refueling techniques

In [325], a machine vision approach has been presented in order to provide a solution for PDR technique. At which, the features are detected and extracted from Hue, Saturation, and Value (HSV) color space images. Then the least square ellipse fitting model is applied to the detected features to find the center of the drogue. From their experiments, it has been shown that the using of HSV color space increases the accuracy of the feature extraction step.

Deng *et al.* developed a system of the BRR technique for the AAR based on stereo vision [85]. At which, the tanker is provided with a binocular camera in order to detect the color characteristics of the markers. Then the system estimates the position of the contacting point of the boom to the receptacle. Although, the system showed good results of the marker detection phase in the outdoor experiments with different light conditions, but also, it needs improvements in the binocular measurements to increase the stability and the accuracy of the pose estimation of the receptacle for the docking phase.

Recently, a visual framework for AAR has been presented in [331]. At which, two classifiers have been combined for the detection and tracking of the drogue. The *D-classifier* is used to detect the drogue from the input images. In addition, the *T-classifier* is used to track the detected drogue. Although the results showed better performance, the system has a limitation in the time of computation which is not suitable for real-time operations.

2.1.4 Autonomous Inspection

Aerial Inspection is one of the most recent and in demand applications, that takes the advances of the UAVs (especially rotor-crafts). Along with the safety and the decreasing of human risk, UAVs have the advantage of reducing operational costs and time of the inspection tasks. However, it is important to keep the image stability against any kind of maneuver [67]. UAVs can perform inspection tasks in different terrains and situations; such as buildings, bridges [221], wind turbines, power plant boilers [52], power lines [87], and even tunnels.

An integrated visual-inertial SLAM sensor has been proposed in [238], in order to be used with the UAVs, for industrial facilities inspection purposes. This system consists of a stereo camera, Micro-Electro-Mechanical System (MEMS) gyroscopes and accelerometers. The UAV performs autonomous flights following predefined trajectories. The motion of the UAV is mainly estimated by the inertial measurements; then it is refined using the visual information. From the experiments, it has

been shown that the system suffers from a delay between the inertial sensors and the stereo camera. Thus, a calibration process is required. In addition, the results showed a drift error of 10cm in the displacement over time.

Another visual-inertial sensor has been introduced in [246]. At which, a visual-inertial stereo camera is used to estimate the pose of the UAV as well as to build a 3D map of the industrial infrastructures while inspection.

In [26], two visual servoing approaches were presented for the power line inspection purposes. Both approaches dealt with the problem of keeping the UAV with a close and determinate distance to the power lines while performing the inspection. In the first approach, an Image-Based Visual Servoing (IBVS) formulation has been combined with the Linear Quadratic Servo (LQS) in order to improve the control design of the UAV. Whilst in the second approach, the control problem was solved using the Partial Posed Based Visual Servoing (PPBVS) model. As it has been shown from their experiments, the PPBVS is more efficient and more robust than the IBVS. However, PPBVS approach is very sensitive to the calibration errors.

Autonomous UAV for wind turbines inspection has been presented in [132, 286]. First, the Global Navigation Satellite System (GNSS) and altimeter are used for positioning the UAV in a determinate distance from the tower, then the UAV are rotated to face the hub using the visual information. These works are based on Hough Transform to detect the tower, the hub, and the blades. The only difference is in the tracking phase, where in [286], the Kalman filter is used to track the center of the hub, while in [132], the tracking is based on optical flow algorithms, then the mo-

(a) Inspection data (b) Inspection drone

Figure 2.2 Building Inspection [16]

tion direction, velocity and distance of the hub and the blades can be estimated. Finally, the UAV flights in a preprogrammed path in order to perform the inspection task.

Recently, RGB-D vision-based system implemented on a UAV for infrastructure inspection purposes has been presented in [16]. In this system, the *Kinect v2* has been used as the main sensor, and the obtained color-depth information is used in order to detect the defects, and provides accurate millimetric measurements of the cracks in the surfaces under inspection. Furthermore, this information is used as a feedback to the control system in order to maintain the autonomous flights. From the results, it has been shown that the system has an overall accuracy of 97% in detecting the wall plane and estimating the distances to the UAV, and 99.1% in measuring the cracks.

2.2 UAVs' Navigation Systems

Modern UAVs aim at higher levels of autonomy with accurate flight stabilization. The main part of an autonomous UAV is the navigation system and its supporting subsystems. The navigation system utilizes information from various sensors in order to estimate the pose of the UAV, in terms of positions (x, y, z) and orientations (ϕ, θ, ψ). Other supporting systems solve relevant tasks; such as obstacles detection and tracking (static or dynamic), or obstacle avoidance.

With this increase in the levels of autonomy and flight stabilization, robust and efficient navigation systems are required. Computer vision algorithms by means of monocular cameras can be helpful to enhance the navigation activities. As it is shown in Table 2.2, the navigation systems are divided into three main subsystems: **Pose estimation** which aims to estimate the position and the attitude of the UAV in two and three dimensional representations, **Obstacle detection and avoidance** that detects and feeds back the position of the obstacles that are situated in the path of the UAV, and finally, the **Visual servoing** subsystem, at which, the maneuver commands are managed and sent in order to maintain the flight stability and following the path. The following subsections (2.3, 2.4 and 2.5) address these three navigation subsystems.

Table 2.2 UAVs vision-based algorithms: Pose estimation, Obstacle detection and avoidance and Visual servoing

System	Description	Method	Related Work
Pose Estimation (Localization)	Estimate the 2D/3D position and orientation of the UAV	Visual Odometry	[44, 70, 86, 94, 96, 104, 120, 169, 187, 229, 239, 245, 257, 313, 319, 345, 346]
		SLAM	[9, 45, 46, 82, 95, 105, 157, 183, 209, 219, 224, 316, 344, 348]
Obstacle Detection and Avoidance	Detect possible obstacles and collision zones and making the suitable avoidance decisions	Stereo	[50, 61, 71, 107, 109, 134, 135, 145, 204, 232, 242]
		Monocular	[7, 15, 43, 179, 200, 202, 228, 261]
Visual Servoing	Maintain UAV stability and flying maneuvers based on visual data		[47, 173, 175, 198, 237, 243, 275, 347]

2.3 Pose Estimation

Pose estimation is the process of estimating the position and the orientation of the vehicle during the motion, based on the information generated by one or more sensors; such as IMU, GPS, vision, laser, ultrasonic, etc. The information can be generated by each sensor separately or by fusing the data from different sensors. Pose estimation is considered as a fundamental phase for any navigation or mapping processes.

2.3.1 Global Positioning System (GPS)

Global Positioning System (GPS) [152, 354] or the Satellite-based Navigation System (SNS) is considered as one of the most known approaches that is used with UGVs [2, 20, 315, 326, 335], UAVs [31, 66, 108, 139, 142, 160] or even Autonomous

Underwater Vehicles (AUVs) [176, 217, 295] in order to provide the 3D position for navigation purposes.

In most cases, the GPS is used as the main sensor for localization process to obtain the position of the vehicles. One of the earlier works that is based on the GPS for localization with UAVs was presented by Hui *et al.* [139]. In this work, the authors showed the effect of using the Differential Global Positioning System (DGPS) to reduce the errors (satellite clock error, satellite position error and delay error) comparing to the use of the GPS receiver alone. Similarly in [66], a DGPS is implemented to a single antenna receiver in order to increase the accuracy of the positioning information.

In these systems, the precision depends directly on the number of satellites connected. This number can be insufficient on urban environments due to buildings, forests or mountains that can reduce the satellite visibility. Furthermore, in other scenarios such as indoor flying, GPS loses its efficiency because of the absence of the satellite signals. Therefore, some expensive external localization systems that use multiple camera system; with or without markers are used; such as the VICON motion capture systems [10, 51, 116, 218] in order to capture the motion of the UAV in indoor environments.

2.3.2 GPS-Aided Systems

Although stand-alone GPS is useful to estimate the position of the vehicles, it also generates errors because of the disability to receive satellites signals, or by the jamming of the signals that consequently may lead to lose navigation information.

UAVs need a robust positioning system to avoid catastrophic control actions that can be caused by the errors in position estimation, so that, different approaches are used to solve this problem. One example of these approaches is GPS-aided systems. In these approaches, the gathered data from the GPS are fused with the information that obtained from other sensors, this multi-sensory fusion can be of two sensors [37, 251, 294] or more than two sensors [146, 235, 305, 353].

GPS/INS approach is one of the most popular configurations, at which, the data from the Inertial Navigation System (INS) and the GPS are fused together to compensate the generated errors from both sensors, and to increase the precision of the localization process. In [334], the data from multiple antennas GPS are fused with the information from the on-board INS using linear Kalman filter. However, this algorithm has been implemented to be used with UAVs, although the experiments

have been performed on a ground vehicle.

Similar works were presented to reduce the position error using Extended Kalman Filter (EKF) [35], or employing Kalman-Complementary filtering [342], or by fusion Strap-down Inertial Navigation System (SINS) data wish the GPS [251, 294].

In [305], an Unscented Kalman Filter (UKF) was implemented to fuse the GPS data with the camera information, and the data obtained from the IMU in order to improve the localization process. This fusion showed improvement in the results comparing to the result of each sensor, However the experiments were limited to simulations.

Moreover, in other works [146, 235], an altitude sensor was added to the GPS/INS system in order to improve the reliability and increase the accuracy of the navigation, by enhancing the accuracy of the GPS vertical measurements. But these systems still have inaccurate results especially if the UAV flies in low altitudes, because the barometer is affected by the ground effect and estimated altitudes lower than the actual ones [234].

Another multi-sensor fusion based system for multiple MAVs was introduced in [310]. At which, the data of the GPS are fused with the information from the Identification Friend-or-Foe (IFF) radar system; for localization enchantment using EKF. In the simulations, it has been proved that by using two GPS receivers better information is obtained, rather than a single GPS receiver.

In [142], a GPS localization system is used on Lockheed *Martin'sSamari* MAV. At which, a greedy source seeking algorithm was used in order to track the radio frequency sources; by estimating the angle of arrival to the source, while observing the GPS signal to noise ratio in order to keep the quality of the GPS signal.

Two main drawbacks appeared on these approaches, affecting the localization process. First, the information are still dependent on the external satellite signals. Second, the lack of precision of the IMU measurements. These difficulties favored the apparition of vision-based systems. These novel approaches enhance the localization by means of computer vision-based algorithms.

2.3.3 Vision-Based Systems

Owing to the limitations and drawbacks of the previous systems, vision-based pose estimation approaches have become one of the main topics in the field of intelligent vehicles applications and gain more popularity to be developed for UGVs [262, 270, 296, 346], AUVs [46, 89, 172, 216], and UAVs [102, 165, 192, 196, 333].

Visual pose estimation methods in particular, are based on the information provided by the visual sensors ot the cameras. Regardless to the type of the vehicle and the purpose of the task, different approaches and methods have been proposed. These methods differ on the type of the visual information used; such as horizons detection [91, 122], landmarks tracking [21, 92], or edges detection [163, 311]. Furthermore, they can be differentiated based on the structure of the vision system: it can be monocular [224, 329, 344], binocular [304, 314], trinocular [144, 212], or omnidirectional [22, 158, 271, 309] camera system.

Some of the early experimental works that use visual information in order to estimate the aircraft attitude were presented in [73, 74, 297, 298]. These approaches are based on the skyline segmentation using forward-looking camera. In these approaches, a Bayesian segmentation model with a Hidden Markov Trees (HMT) model was used to identify the horizon based on the color intensities, and texture clues, in order to estimate the *roll* angle or both *roll* and *pitch* angles, as the work presented in [91]. These approaches provide successful results in high altitudes where the process of skyline segmentation is relatively easy. On the other hand, in low altitudes or indoor environments, the possibility to detect the horizon is very low due to the complexity of this kind of environments.

Two famous philosophies have appeared to deal with the vision-based pose estimation problem; Visual Simultaneous Localization And Mapping (VSLAM) and Visual Odometry (VO). The following sections are giving a review of these two topics.

Simultaneous Localization And Mapping (SLAM)

Simultaneous Localization And Mapping (SLAM) algorithms [34, 76, 90], in general aim to construct a consistent map of the environment, and simultaneously estimate the global position of the robot within this map.

Approaches such as those that have been presented in [9, 24, 45], introduced different camera-based algorithms; such as Parallel Tracking and Mapping (PTAM) [164] and Mono-Simultaneous Localization And Mapping (MonoSLAM) [82] in order to perform VSLAM on aerial vehicles.

Blosch *et al.* used a downward-looking camera on the Hummingbird quadcopter based on a vision approach for localization purposes [45]. The UAV pose is estimated using the VSLAM algorithm, and thereafter, a Linear Quadratic Gaussian (LQG) control design coupled with Loop Transfer Recovery (LTR) (LQG/LTR)

are applied to stabilize the vehicle at a desired setpoints.

In [224], a vision-based SLAM with visual expectation algorithm was introduced. In this approach, a place recognition algorithm, based on the patch tracking, is used to estimate the *yaw* angle and the translation speed of the vehicle. In addition, the visual expectation algorithm is used to improve the recall process of the visited places. This is achieved by comparing the current scene with the library of saved templates. Finally, both algorithms are combined to a RatSLAM [225] for constructing the maps. However, this system loses its efficiency with the new scenes that are not visited before by the vehicle.

In [157], a SLAM approach with RGB-D cameras has been presented. In this approach, direct frame-to-frame registration method, with the entropy-based model, were used to reduce the drift error of the global trajectory.

Another direct frame registration method has been presented in [95]. In contrast to the RGB-D approach, this method implemented a monocular SLAM with the advance of the ability to construct large scale maps.

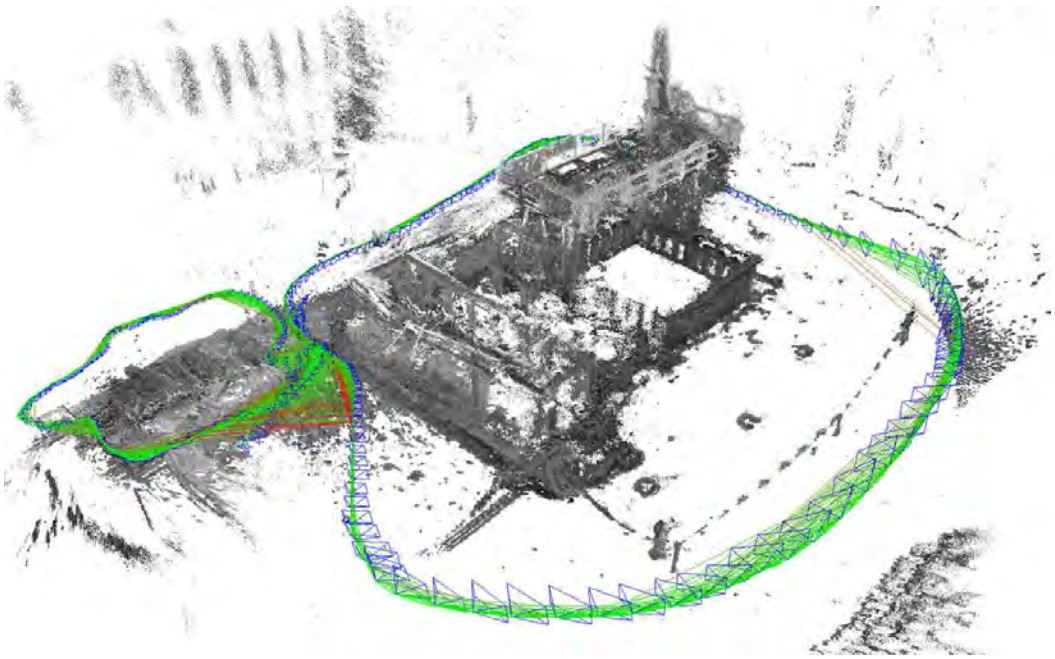


Figure 2.3 Example of LSD-SLAM [95]

A laser-assisted system was presented in [344] to estimate the attitude of the UAV. At which, the pose of the UAV is obtained by a laser scan matching based on the Sum of Gaussian (SoG). The laser spot is captured by a camera mounted on the UAV, and by using gray correlation template matching model, the distance of

the spot is obtained. Thereafter, the pose of the UAV is estimated by using SoG. In addition, EKF is used to combine the inertial information with the visual system in order to improve the navigation process.

Another VSLAM approach was presented in [348] in order to control a nano quadcopter. The motion of the quadcopter has been obtained based on an optical flow model. In addition, to eliminate the drift error in the flight a PTAM was used. Similarly, to the work presented in [344], a Kalman filter was used to fuse the data from the IMU and the barometer, with the visual information, in order to improve the motion estimation. The main drawback of this system is the difficulty to achieve the hover mode for a long time, this is because of the limitation of the optical flow algorithm.

Although SLAM, or in particular VSLAM, is considered to be a precise method for pose estimation purposes, the outliers in the detection affect the consistently of the constructed map. Furthermore, these algorithms are complex and computationally expensive.

Visual Odometry (VO)

Visual Odometry (VO) algorithms [240, 269] handle the problem of estimating the 3D position and orientation of the vehicle. The estimation process performs sequential analysis (frame after frame) of the captured scene in order to recover the pose of the vehicle. Similar to VSLAM, this visual information can be gathered using monocular cameras [124, 256, 257, 308] or multiple cameras systems [203, 229, 313].

In contrast to VSLAM, VO algorithms deal to estimate consistent local trajectories, in each instant of time without maintaining all the previous poses.

VO firstly proposed by Nistér [240, 241], it was inspired by the traditional wheel odometry, to estimate the motion of ground vehicles, using stereo camera, incrementally by detecting the Harris corners [127] in each frame. In this approach, the image features are matched between two frames, and linked into image trajectories, by implementing a full structure-from-motion algorithm that takes advantage of the 5-point algorithm and RANdom SAmple Consensus (RANSAC) [101]. From his experiments, it was proved that the VO accuracy is better than the wheel odometry, with position error of [0.1% to 3%] of the total trajectory.

Within the NASA Mars Exploration Program (MER) [65, 203], a stereo VO algorithm based also on Harris corner detector has been implemented on the MER rover; to estimate its 3D pose in the terrain Mars (feature-poor terrain). Related

works to employ VO algorithms on the ground vehicles have been presented in [100, 189, 270, 283].

A hybrid model of visual-wheel odometry is presented in [346]. In this model, the position of the ground vehicle is estimated based mainly on a monocular camera, then both of the rotation and the translation are recovered separately using the Ackermann steering model.

Recently, different motion estimation schemes based on stereo VO algorithms are presented to be applied on the UAVs; such as the works in [104, 245, 313].

In [313], stereo VO system is presented to enhance initializing the pose of the stereo camera. At which, the initializing process is based on a sequence of 8-10 frames, instead of using a single pair. Although this system showed good results in large-scale environments, it cannot be used with the MAVs because of the requirement of a big size stereo camera with a baseline of 0.78m.

On the other hand, in [104, 245], a new small size RGB-D visual-inertial sensor [239] has been used on the MAVs. The first work used a probabilistic model to incorporate the RGB-D camera with the IMU, in order to estimate the motion of the UAV, and build 3D models of the environment. The later work, presented a stereo VO algorithm based on feature tracking technique, where, a combination of Features from Accelerated Segment Test (FAST) [258] and Binary Robust Independent Elementary Features (BRIEF) [55] are used for the feature tracking step. However, this combination provides fast processing, it cannot provide accurate data compared to other algorithms such as SIFT [194].

Approaches based on optical flow techniques have been presented in order to estimate the motion of the UAV [182, 227, 287].

In [287], an optical flow method has been introduced for autonomous navigation purposes. At which, the UAV motion is estimated based on the stereo information. In this system, two *fisheye* cameras have been mounted back to back, in order to generate the stereo data that are required for the optical flow. However, there experiments were limited to a simple shape (square shape). Furthermore, the obtained results represented an average error up to 1.7% of the total path, which is relatively big with respect to the short distance path (10×10 meters).

More *et al.* presented an optical flow-based visual odometry [227]. This system applies the Lucas-Kanade method [195] using monocular camera in order to estimate the 2D UAV motion. However, this system was verified in a laboratory setup using a fixed rig platform.

Furthermore, although the stereo camera used in these systems is small and lightweight, suitable to mount on a small UAVs, the small baseline caused a significant limitation of the system in the large-scale environments.

The research in [120], introduced an optical flow vision-based system, combined with the on-board IMU to estimate the motion of the UAV. In this system, the Shi-Tomas algorithm [278] is used for the feature detection, then the pyramidal LK model [48] is used to track the detected feature. Thereafter, the obtained velocity from IMU is used to compensate the velocity error estimated by the optical flow algorithm. The same concept of using the combination of the optical flow and IMU model is presented in [187]; for controlling the hover flight mode of the quadcopter. The main limitation of this model, is providing an unbalanced representation of the scene, when there are insufficient number or features, or if the tracked features are not distributed across the image plane.

Another vision-IMU navigation system has been presented in [308]. In the vision-based estimation, SIFT is applied in order detect and track the feature points, and the 2D UAV motion is estimated from the planar homographies between the tracked feature points. In addition, an EKF is applied to fuse the visual and IMU data in order to obtain the optimal estimation of the UAV pose. The experimental results show that the system has a standard deviation in the position estimation of 8.87 for the vision-based method, and 1.16 for the vision-IMU combination.

A position estimation approach of aerial vehicles, based on line detection and corner extraction is presented in [150]. In which, lines and corners are extracted by Hough transform and Harris corners detection, then the rotation, translation and scale are estimated. Finally, a geometric model estimation is used to map the high-resolution image onto a low-resolution, providing the position estimation.

A monocular camera-based navigation system for an autonomous quadcopter was presented in [169]; to determine only the UAV *yaw* and vertical speed. One of the limitation of this method is that the UAV can only operate along paths it has traveled during a human-guided training run. Moreover, these paths can be composed only from straight-line segments with a limited length.

In [264], an approach of vision-based (2D-3D) pose estimation of UAVs have been presented. An which, the algorithm aligns 2D data from the aerial image into a geo-referenced ortho satellite image 3D based on fuzzy reasoning system.

An euclidean homography method was presented in [149]; to maintain the vehicle navigation, when GPS signals are not available. This system allows sets of

feature points of a series of daisy-chained images to be related; such that the position and orientation can continuously be estimated. However, this method was limited to simulation results, and the disability to estimate the depth where there is a change in environment planes.

Similarly in [201], a vision-aided navigation system is used to replace the GPS when it is temporarily denied. A single camera system detects, tracks, and geolocates 3D landmarks observed in the images; in order to estimate the absolute position and velocity data.

Another multi-sensor data fusion model is introduced in [263]. In which, the system uses an EKF to fuse the vision information which provides attitude and position observations, with the data from the IMU motion model, for accurately determining the pose parameters of the vehicle.

In this dissertation, a vision-based pose estimation approach is proposed. This visual system estimates the 3D attitude of the UAV based on the image sequence from a monocular camera. The obtained information is used in the localization phase, and for the vision-based control tasks; such as hover maneuver, waypoint tracking and obstacle avoidance.

2.4 Visual Obstacle Detection and Avoidance

Obstacle detection and avoidance is a fundamental phase in any autonomous navigation system. In addition, this process is considered as a challenging process, especially for vision-based systems.

In vision-based navigation systems, different approaches were presented to solve the problem of obstacle detection and avoidance. Approaches such as [42, 50, 109, 135, 232], built a 3D model of the obstacle in the environment. Other works calculate the depth (distance) of the obstacles; such as in [145] and [261].

A technique based on stereo cameras; in order to estimate the proximity of the obstacles, was introduced [204]. At which, the system detects the size and the position of the obstacles based on the disparity images and the view angle. Furthermore, this technique calculates the relation of the size and the distance of the detected obstacle to the UAV.

Another stereo vision-based obstacle detection for ground vehicles is presented in [50]. At which, a Voxel map is reconstructed from the 3D point cloud provided by the stereo camera. Thereafter, a linear Kalman filter is used to distinguish between

the moving and stationery obstacles. Finally, with the aid of the computed ego-motion, the system estimates the position and the velocity of the detected obstacles.

All these approaches have the disadvantage of the high cost in the computation time.

Whilst bio-inspired (insect, animal or human like) approaches estimates the presence of the obstacle efficiently, without calculating the 3D model; such as using motion parallax (optical flow) [42, 136, 220] or perspective cues [43, 60, 63]. However, optical flow approaches cannot identify the forward movement, due to the aperture problem, thus frontal obstacles would provide only movement component normal to the detected edges in the image, not providing frontal movement information *per se*. Perspective cues approaches worked well in the structured environments [228].

In [84], it was presented an approach based on the texture and color variation cue; to detect obstacles for indoor environments. However, this approach works only with detailed textures. Furthermore, their experiments were limited to indoor environments.

Working with Hybrid MAVs, Green *et al.* proposed an optical flow approach, mimicking the biological flying insects, by using dual cameras mounted on a fixed-wing UAV, in order to detect and avoid lateral obstacles [121]. Besides the detection of lateral obstacles only, some limitations appeared in avoiding large obstacles like walls. In addition, from the experiments, the avoidance algorithm is insufficient if the UAV flies in a straight path.

In [177], SIFT descriptor and Multi-scale Oriented-Patches (MOPS) are combined to show the 3D information of the obstacles. At which, the edges and the corners of the object are extracted using MOPS by obtaining and matching the MOPS feature points of the corners, then the 3D spatial information of the MOPS points is extracted. After that, SIFT is used to detect the internal outline information. However, the presented approach has expensive computational time (577ms).

Bills *et al.* [43] proposed an approach for indoor environments with a uniform structure characteristics. In this work, Hough Transform is used to detect the edges that are used; to classify the essence of the scene based on a trained classifier. However, their experiments were limited to corridors and stairs areas.

A saliency method based on Discrete Cosine Transform (DCT) is presented in [200] for obstacle detection purposes. From the input images, the system assumes that the obstacle is a unique content in a repeated redundant background, then

by applying amplitude spectrum suppression, the method can remove the background. Finally, by using the Inverse Discrete Cosine Transform (IDCT) and a threshold algorithm, the center of the obstacle is obtained. Furthermore, a pinhole camera model is used to estimate the relative angle between the UAV and the obstacle, this angle is used with a PD controller to control the heading of the UAV for obstacle avoidance.

In [261], the authors presented an approach for measuring the relative distance to the obstacle. At which, the camera position is estimated based on the EKF and the gathered data from the IMU. Then the 3D position of the obstacle can be calculated by back projecting the detected features of the obstacle from its images.

An expansion segmentation method was presented in [53], in which a conditional Markov Random Field (MRF) is used to distinguish if the frontal object may represent a collision or not. Additionally, an inertial system is used to estimate the collision time. However, the experiments of this work was limited to simulations.

Another approach presented in [228], used the feature detection algorithm in conjunction with the template matching; to detect the size expansions of the obstacles. However, the experiments were limited on a tree-like obstacles and did not show results of other shapes.

In [98], an optical flow based system has been presented to detect the obstacles and junctions in outdoor environments. This system is based on the *Horn & Schunk* method [133]; in order to look for the collision free areas and the junctions in a predefined flight path. In addition, a PID controller is used as a low-level control scheme. However, all the experiments were limited to virtual flights in Google Earth software.

Kim *et al.* presented a block-based motion estimation approach for detecting the moving obstacles (humans) [161]. In which, the input image is divided into smaller blocks, then the system compare the motion in each block through consecutive images. However, their experiments were limited in detecting large size obstacles (humans) in indoor environments.

In addition, surveys of different approaches of UAVs guidance, navigation and collision avoidance methods and technologies are presented in [18, 155, 337]. Recently, Mcfadyen *et al.* presented a literature review of the vision-based collision avoidance systems [215].

Detecting and avoiding frontal obstacles using monocular camera is considered as a challenging problem because of the absence of the optical flow or the motion

parallax. However, size expansion provides useful information for detecting the obstacles that are moving towards the UAV.

From the bio-inspired point of view, the human visual system has the ability to extract information correctly of the objects that are moving toward them [280]. In addition, Gibson illustrated the ability of the human visual system to identify the approaching of the objects related to the expansion of its size, by both eyes or even one eye [114].

From this aspect, this dissertation presents a bio-inspired approach using a monocular camera in order to mimic the human behavior of obstacle detection and avoidance applied on UAVs.

2.5 Visual Servoing

Visual Servoing (VS) is the process of using the information that are obtained by the visual sensors as a feedback in the vehicle (UAV) control system. Different inner-loop control systems have been employed in order to achieve the stabilization of the UAVs; such as PID [115, 148, 210], optimal control [291], sliding mode [175], fuzzy logic [188], and cascade control structure [41]. References such as [38, 93, 306] provide detailed information about the principles and theories related to the UAV flight controlling systems. On the other hand, higher level control systems can be used for guidance purposes; such as waypoints tracking or path following [93, 244].

A comparative study has been introduced in [19] to evaluate two controllers (*mode-based feedback linearizing* and *backstepping-like* control) using visual feedback. At which, an external camera and the on-board gyroscopes are used to estimate the UAV angles and position. From the simulations, it has been found that the backstopping controller is better than feedback stabilization.

In [175], an image-based visual servoing has been described; to use the 2D information as an input to the adaptive sliding mode controller for autonomous landing on a moving platform.

A visual system based on two cameras (external camera located on the ground and on-board camera), was presented in [226] for flight stabilization purposes in the hover modes. At which, both cameras are set to see each other, and a tracking algorithm is used to track the color blobs that are attached to the cameras. Thereafter, the pose of the UAV is estimated. Finally, the Linear Quadratic Tracking (LQT) controller and optimal LQG control are used with the visual feedback in order to

stabilize the attitude of a UAV. However, the performance of the proposed controller was verified in simulations.

A design of fuzzy control for tracking and landing on a helipad has been presented in [243]. In this approach, four fuzzy controllers are implemented to control the longitudinal, lateral, vertical, and heading velocities to keep the UAV in the center of the moving helipad. However their experiments were limited to indoor environments. In addition, the estimation of the UAV pose is based on a vision algorithm using the implementation of the argument reality codes.

A combination of the classic and multi-objective genetic-based fuzzy controllers have been proposed in order to improve the robustness, and the time response of controlling the altitude hold mode [28]. However, the results were obtained from simulation without performing real flights.

An inertial-visual aided control system was presented in [33]. The Kanade-Lucas-Thomasi (KLT) feature tracker algorithm is used to estimate the UAV attitude, then the values are sent to a PID control system. However, this control system is lacking of a filtering, resulting a significant drift error.

Recently, Lyu *et al.* proposed a visual servoing system that is based on cooperative mapping control framework of multiple UAVs [198]. This framework consists of a master UAV which leads and controls multiple slave UAVs. Both master and slaves are equipped with downward-looking cameras to obtain rectified images of the ground. The visual servoing is achieved by using the moment of the SIFT features. Where the extracted SIFT features by the master UAV are matched with the features extracted by the slave UAVs. Afterwards, the moment feature is generated. Finally, based on the obtained information, a visual servoing controller is applied to guide the slave UAVs to follow the master UAV. However, all the results are obtained by simulations.

2.6 Summary

In this chapter, complete review of vision-based systems for UAVs have been reviewed as a whole methodology to cope with cutting-edge UAV technology, where environment perception has been studied as complex and essential task for UAV navigation and obstacle detection and avoidance in the last decade. The advantages and improvements of computer vision algorithms towards the presented reliable solutions have been presented through real results under demanding circumstances, such as, pose estimation or aerial obstacle avoidance. So, complex tasks and applications have been analyzed and difficulties have been highlighted, where the trustable performance of the vision-based solutions and the improvements in relation to the previous works of the literature are provided.

The different vision-based systems mounted in an UAV represent actual applications and help to overcome classical problems, like autonomous landing or autonomous inspection, among others. So, the strengths of the presented computer vision algorithms for UAVs have been clearly stated in this chapter. However, presented applications have specific drawbacks that should be taken into account. That is, the vision-based systems are low cost sensor devices, which provides high amount of information, but have the drawback of the high sensitivity to lighting conditions (e.g. direct sun light may lead to lack of information). Moreover, all the presented algorithms and applications give full understanding and convergence to the next generation of UAVs.

The presented survey provides a full review of the vision-based advance in literature in the last decade. providing full understanding of novel applications derived from them, and fostering for the development of outstanding UAVs, capable of the most advanced and modern tasks in the most challenging scenarios

CHAPTER 3

Monocular Vision-based Pose Estimation

This chapter[1] presents a robust pose estimation solution for estimating the six Degrees of Freedom (6 DOF) attitude of the UAV in real-time flights, using downward-looking monocular on-board camera.

The solution of estimation the UAV pose is addressed as a feature-based method. At which, the pose estimation problem is solved based on the decomposition of the *world-to-frame* and *frame-to-frame* homographies, that are calculated from the ex-

[1] Publications of the author related to the chapter:

- SIFT and SURF Performance Evaluation and the Effect of FREAK Descriptor in the Context of Visual Odometry for Unmanned Aerial Vehicles [11].
- Indoor and Outdoor Navigational System for UAVs based on Monocular Onboard Camera [12].

tracted and tracked feature points from the consecutive frames that are captures during the flight.

3.1 Pose Estimation

The pose estimation strategy of an aerial vehicle is shown in Figure 3.1, at which the UAV captures new frame consecutively each time. Thereafter, the system estimates the 3D motion information (position (x, y, z) and orientation (ϕ, θ, ψ)), by matching and tracking the extracted feature points between these consecutive frames, using a single camera mounted downward-looking on the aerial vehicle, as it is explained in Algorithm 1.

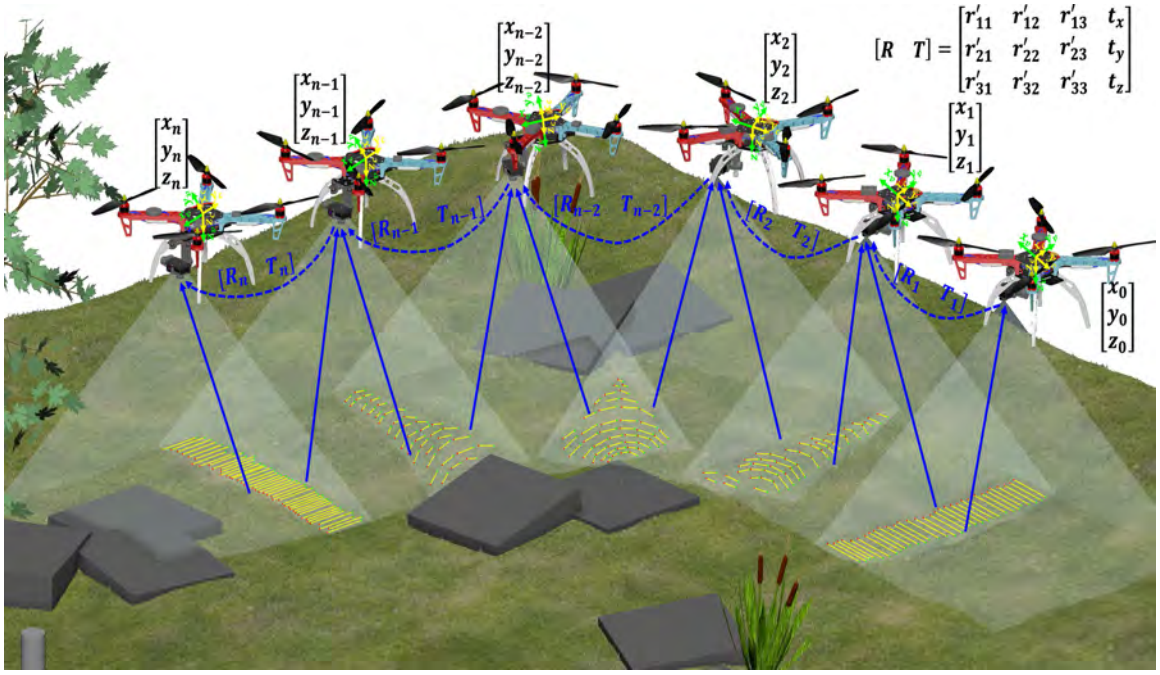


Figure 3.1 Pose estimation strategy for UAVs: the motion from point to point is based on the rotation and translation matrices, represented by R and T respectively

Feature points detection and description play very important role in many of computer vision applications, specifically in robot visual navigation systems; such as visual odometry or VSLAM, which need reliable high speed processing algorithms with low memory load.

Based on the algorithm, in each frame, high number of reliable features points should be detected and matched in order to extract useful motion information, and consequently to estimate accurate data about the UAV position and orientation.

Algorithm 1: Homography-Based Pose Estimation

Input: Input frames F, Intrinsic matrix C

Output: Pose vector $POSE = (x, y, z, \phi, \theta, \psi)$, Velocity vector $V = (u, v, w)$

Define: Current frame F_i, Previous frame F_{i-1}, Current keypoints $KP_i[\,]$, Previous keypoints $KP_{i-1}[\,]$, Number of keypoints (N, M), $n \in N$, $m \in M$, $n = m$, Matched points q, Distance ratio *thresh*, Homography H, Rotation matrix R, Translation vector T, Time Δt

begin

  while *isFlying( )* **do**

    ${}^{0}\mathrm{H}_{w} \leftarrow$ clcInitialHomography( ) `// Estimate world-to-frame homography`

    $F_{i-1} \leftarrow$ getNewFrame( )

    $F_{i-1} \leftarrow$ CorrectIllumin(F_{i-1}) `// Correct image illumination`

    $F_{i-1} \leftarrow$ ImageUndistortion(F_{i-1}) `// Correct distorted images`

    if *Navigation*$\leftarrow$*isActivated( )* **then**

      $F_i \leftarrow$ getNewFrame( )

      $F_i \leftarrow$ CorrectIllumin(F_i)

      $F_{i-1} \leftarrow$ ImageUndistortion(F_{i-1})

      ($KP_{i-1}(N)$, $KP_i(M)$) $\leftarrow$ Detect_DescribeKeypoints(F_{i-1}, F_i)

      ($q_{i-1}(n)$, $q_i(m)$) $\leftarrow$ MatchSymKeypoints($KP_{i-1}(N)$, $KP_i(M)$, *thresh*)

      if *isEquiPixDistance(50%*($q_{i-1}(n)$, $q_i(m)$)*)* **then**

        ${}^{i}\mathrm{H}_{i-1} \leftarrow$ clcFram2FramHom($q_{i-1}(n)$, $q_i(m)$) `// Estimate frame-to-frame homography`

        ${}^{i}\mathrm{H}_{w} \leftarrow$ clcFinalHom(${}^{i}\mathrm{H}_{i-1}$, ${}^{0}\mathrm{H}_{w}$) `// Estimate final homography`

        $[R, T]_i \leftarrow$ decomposeHom(${}^{i}\mathrm{H}_{w}$, C)

        $POSE_i \leftarrow$ Estimete_Pose(R, T)

        $V_i \leftarrow$ getVelocities(T, Δt)

        ${}^{0}\mathrm{H}_{w} \leftarrow {}^{i}\mathrm{H}_{w}$

      end

      $POSE_i = POSE_{i-1}$

      $V = 0$

    end

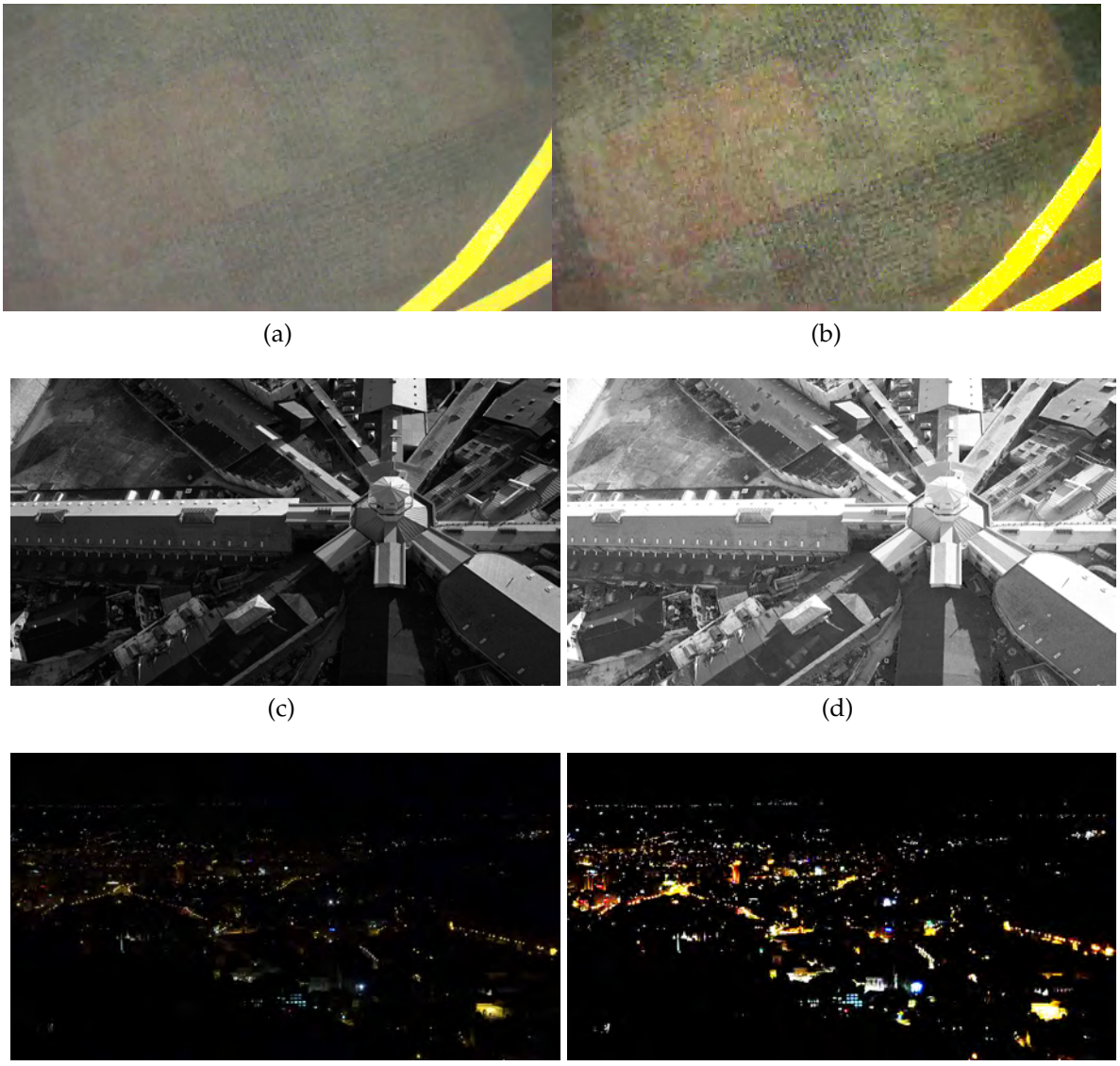
    $F_{i-1} \leftarrow F_i$

  end

end

3.2 Illumination Correction

Visual Sensors usually work with the exposure as well as the contrast properties; in order to deal with the illumination effects in the environment. The lens aperture and the shutter speed in the camera control the exposure (the amount of light that falls into the image sensor), which consequently affects the image to be brighter with shorter exposure time, or darker with longer exposure time. This exposure, in turn, affects the contrast and the sharpness of the image.



(a) (b) (c) (d) (e) (f)

Figure 3.2 Example of illumination corrected images.

Therefore, as it is illustrated in Figure 3.2, it is important to correct the illumination of the captured images before performing any processing. In order to achieve this, the first step is to expand the dynamic range of the image intensities; for that, the histogram equalization technique is used, where the equalized histogram is calculated as follows:

$$I_{equhist} = \frac{L(cdf_i - cdf_{min}) - (N \times M)}{N \times M} \tag{3.1}$$

where $I_{equhist}$ is the equalized image, L is the number of gray levels, cdf_i is the cumulative frequency of the histogram array $hist$, where $cdf_i = cdf_{i-1} + hist_i$, cdf_{min} is the minimum non-zero value of the cumulative distribution function, and (N, M) are the images dimensions.

The second step is to control the overall brightness of the equalized image. For this purpose, *Gamma Correction* method is used. Gamma correction is defined by using the Power Law Transform as follows:

$$I_{cor} = I_{equhist}^{\frac{1}{\gamma}} \tag{3.2}$$

where γ is the gamma value, in which the image tends to be darker if $\gamma < 1$ and appears brighter when $\gamma > 1$.

3.3 Feature Detection and Matching

During the last decade, a variety of feature detectors; such as SIFT, SURF [36], FAST, STAR [6], Binary Robust Invariant Scalable Keypoints (BRISK) [180]), and descriptors; such as SIFT, SURF, BRIEF, ORB, FREAK [17]) are proposed and applied to visual navigation purposes. For real-time navigation applications, the performance and the robustness of the detection and description processes are required; therefore, many surveys and comparisons of different feature point detectors and descriptors are presented.

In [300], a survey with a comparison of many detectors and descriptors was reported, and as a conclusion the SIFT and SURF have the same accuracy and robustness. However, the SURF is more efficient. Another comparison of affine region detectors was presented in [223], at which the authors concluded that the SIFT has the best results, excluding SURF from the comparison.

A comparison of FREAK and SURF descriptors in the context of pedestrian de-

tection was reported in [272]. In this study, the author concluded that SURF is more robust than FREAK for pedestrian detection issued.

In [273], an experimental study using indoor mobile robot was reported, and the authors concluded that the FAST-BRIEF pair is a good choice when processing speed is a concern. However, SIFT was excluded from the study.

Due to flying in unknown environment structures, the captured frames are affected by different conditions; such as the illumination variation which may induce to noise and error. However, the feature points need to be extracted accurately even under these conditions.

This chapter focuses on SIFT and SURF algorithms; due to that both algorithms provide an invariant detection method. In other words, their ability to identify and localize accurately the feature points, even under different image conditions; such as scale, rotation, illumination or image noise. Additionally, they present results with high level of robustness.

Therefore, a performance evaluation between SIFT and SURF detection and description methods is presented using datasets of several images under typical image transformations (noise, scale and rotation). In addition, studying the effect of pairing FREAK descriptor with SIFT and SURF detectors to be used for real-time visual navigation purposes in aerial vehicles.

The term of feature detection refers to the process of identifying an image feature point (*keypoint*), which differs from its nearest neighbors in term of texture, color or intensity. Whilst, feature description is the process of extracting a local patch around the detected keypoint to be compared with other features.

The three methods of feature detection and description that are used in the performance evaluation study are:

- **SIFT:** Scale Invariant Feature Transform was presented as an algorithm for extracting the feature points from images (Figure 3.3). These features can be invariant in orientation and scale.

  SIFT is based on four major steps of computation: *Scale-space extrema detection* (which uses the Difference of Gaussian (DOG) in order to identify the Keypoints from a pyramid of scales), *Keypoint localization*, *Orientation assignment* and *Keypoint descriptor*.

- **SURF:** Speed-Up Robust Features is an orientation and scale invariant detector and descriptor, inspired by SIFT detector and descriptor (Figure 3.4). This

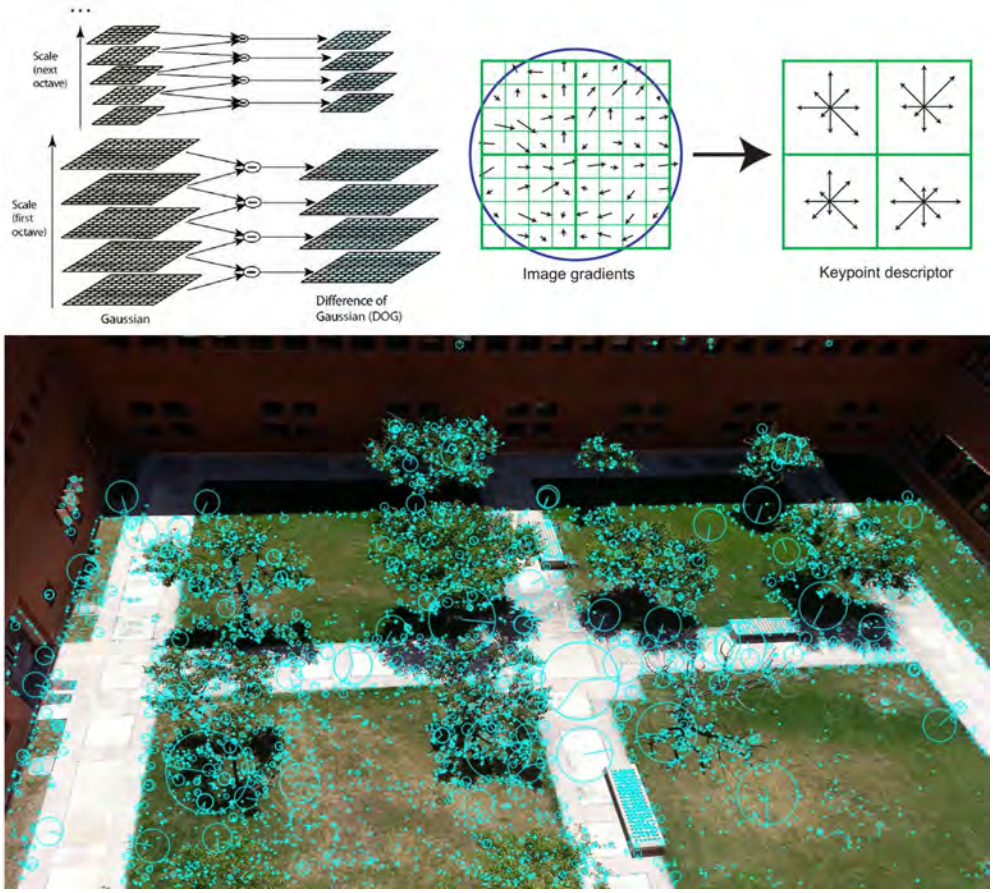


Figure 3.3 SIFT Algorithm.

method takes the advantage of the integral images to gain the speed in the processing.

Unlike SIFT method, SURF method is based on two major steps: *Keypoint detection* (which uses a Laplacian of Gaussian (LOG) on the images, then the determinants of the Hessian matrix are used to identify the Keypoints) and *Keypoint description*.

- **FREAK:** Fast Retina Keypoint descriptor concept is adapted from the biological human visual system (retina), at which, the system computes a cascade of binary strings, by comparing the intensities of the given image using a circular retinal sampling grid. This descriptor is faster to compute with lower memory load.

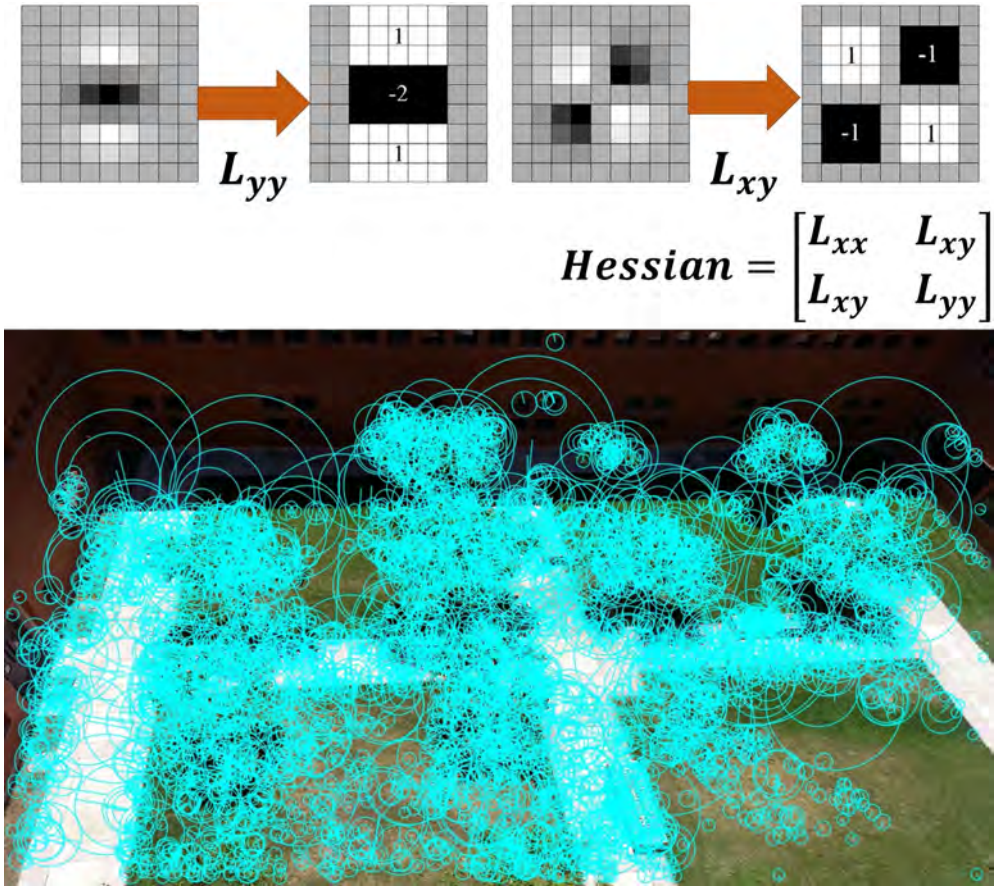


Figure 3.4 SURF Algorithm.

3.3.1 Datasets

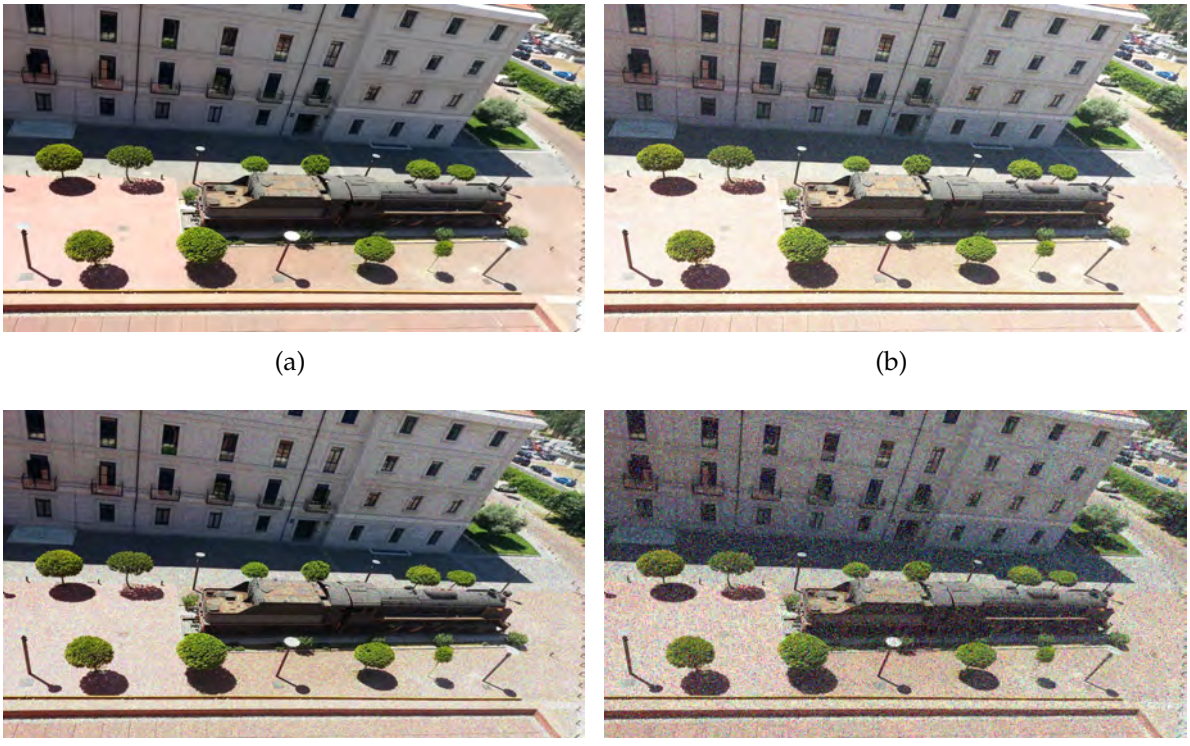
In this evaluation study, all the detection and description algorithms have been verified for image matching using three datasets of a total number of 27500 high-resolution images, with size 1270 $\times$ 720 pixels. The images are captured by the quadcopter flying in outdoor and indoor environments, under various illumination conditions. These datasets are divided into three groups:

- **First Group:** Consists of 8000 images that are captured form outdoor flight with total distance of 61.1m.
- **Second Group:** Consists of 8500 images that are captured from outdoor flight with total distance of 78.6m.
- **Third Group:** Consists of 11000 images that are captured from indoor flight with total distance of 153.6m.

3.3.2 Image Transformations

The performance and robustness of each method in this study are evaluated against different image transformations:

- **Noise Invariance:** To evaluate the noise invariance, three types of noise are applied on the images:
  - Gaussian white noise with $\sigma^2 = 0.1$.
  - Salt-and-Pepper with density of 20%.
  - Multiplicative white (Speckle) noise with zero mean and $\sigma^2 = 0.04$.



(a) (b)

(c) (d)

Figure 3.5 Noise effect: **a:** Reference image, **b:** Gaussian, **c:** Salt-and-Pepper and **d:** Multiplicative white (Speckle).

- **Rotation Invariance:** To evaluate rotation invariance, the test images are rotated at different angles (15° and 30°) in anti-clockwise direction.
- **Scale Invariance:** To evaluate scale invariance, the test images are resized to 50% of the reference images size.

(a) (b)

Figure 3.6 Rotation effect: **a:** 15° and **b:** 30°.

(a) (b)

Figure 3.7 Scale effect: **a:** reference image, **b:** 50% scaled image.

3.3.3 Evaluation

Using the original images as reference, the performance of the detectors and descriptors is evaluated by studying the matches from the extracted points from the reference and the transformed images. The feature points are detected and the descriptors of each point are extracted in both images (reference and transformed). Thereafter, Brute-Force algorithm is applied to each descriptor in the transformed image, in order to match the detected feature points with their correspondings in the reference image, using a distance threshold. If the distance is less than or equal the threshold, the correspondent feature is returned.

Generally, the evaluation is focused on three criteria; **Repeatability:** the percent-

age of the features detected on the scene in both images, **Accuracy:** the localization of the detected features and **Efficiency:** the detection and description should be a time-critical process.

To study these criteria, the following standards have been taken into consideration:

- **Speed per frame:** The absolute total time required to the feature detection and description of a single frame.

- **Speed per feature:** The time required of the detection and description process for single feature; which is calculated as follows:

$$t_f = \frac{T}{N} \tag{3.3}$$

  where, T is the total time divided and N is the number of features.

- **Percentage of matched features:** The ratio of the successfully matched features of the transformed image to the reference image.

- **Average detection error:** The average distance of the feature position in the reference and transformed image. Large values indicate high number of false positives and less accuracy of the detector.

- **Features count deviation:** To estimate how slight exposure changes affect feature detection, and is calculated as follows:

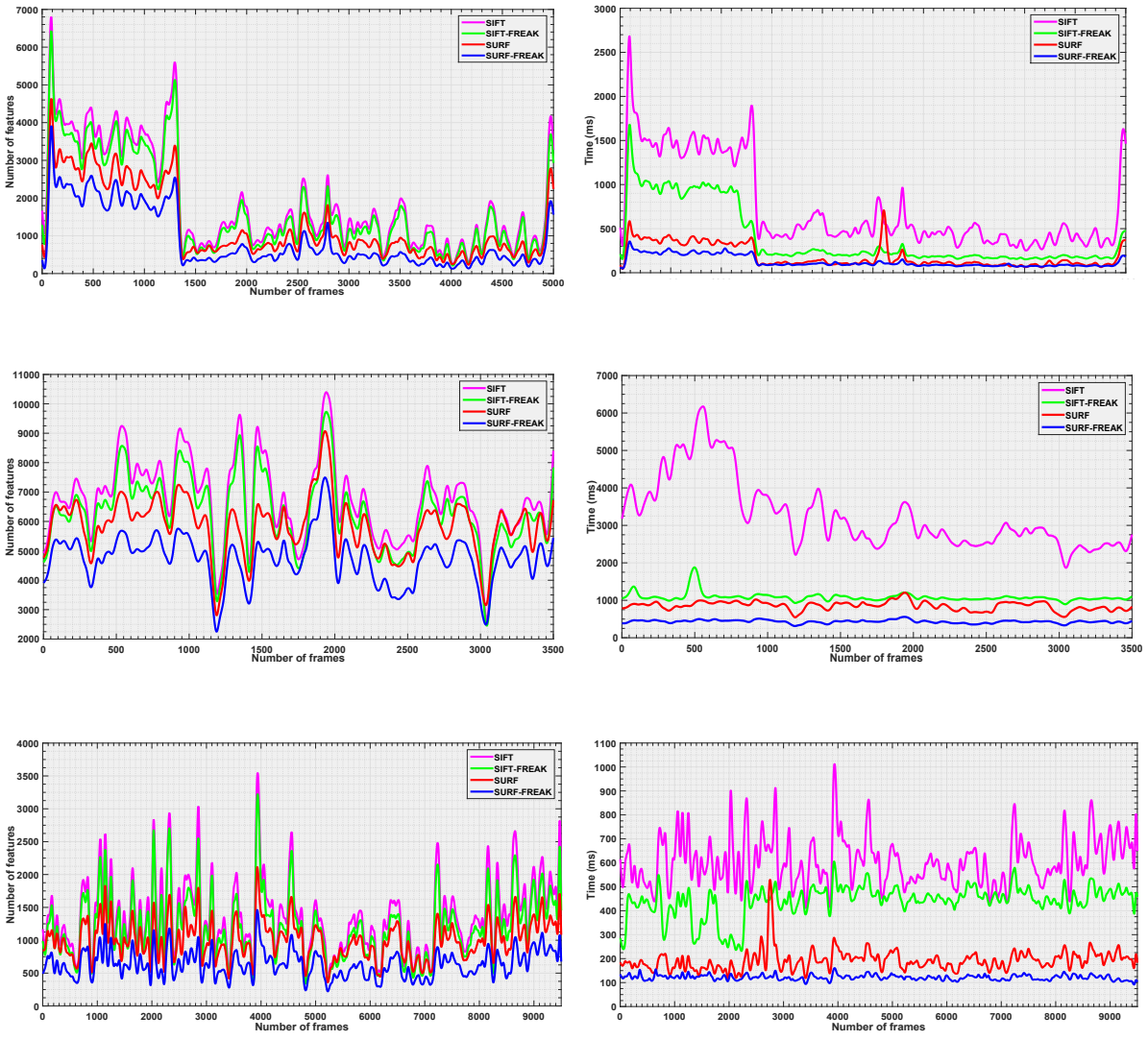
$$\tau = \frac{1}{N} \sum_{i=0}^{N} \frac{x_{o_i} - x_{t_i}}{x_{o_i}} \tag{3.4}$$

  where, x_{o_i} and x_{t_i} are the number of features from the reference and the transformed frames respectively.

On the other hand, the matched feature points by the four methods are used as inputs in the pose estimation algorithm; in order to estimate the UAV position (x, y, z) and orientation (ϕ, θ, ψ). Thereafter, the four generated trajectories are compared to a predefined ground truth and DGPS data; to estimate the accuracy and computation time of each one.

Reference images

In order to evaluate the performance of the presented methods, the feature points are extracted from the reference images, and then, the number of detected features and the computation time of the detection step are calculated in each method. From experiments, as it is presented in Figures 3.8a, 3.8c, and 3.8e and it is shown in Table 3.1, that the classical SIFT (detector and descriptor) detects the highest number of feature points, followed by SIFT-FREAK combination, and finally SURF and SURF-FREAK have less number of the feature points.



(e) (f)

Figure 3.8 Reference images; **a, c, e:** feature points and **b, d, f:** Computational time of 1^{st}, 2^{nd}, 3^{rd} dataset respectively.

Table 3.1 Reference Images

	Dataset	SIFT	SURF	SIFT-FREAK	SURF-FREAK
Number of features	**G1**	1877	1047	1695	885
	G2	6743	5840	6269	433
	G3	1359	1000	1200	635
Time (ms)	**G1**	738.4	184.4	392.2	126.8
	G2	3262.1	853.8	1083.9	432.4
	G3	606.6	192.0	440.8	121.6

However, the SIFT has been found as the slowest in terms of computation time, while SIFT-FREAK has reduced the time of computing significantly, approximately 50% of the classical SIFT. The comparison of the computation time for those four algorithms is shown in Figures 3.8b, 3.8d, and 3.8f.

Rotation invariance

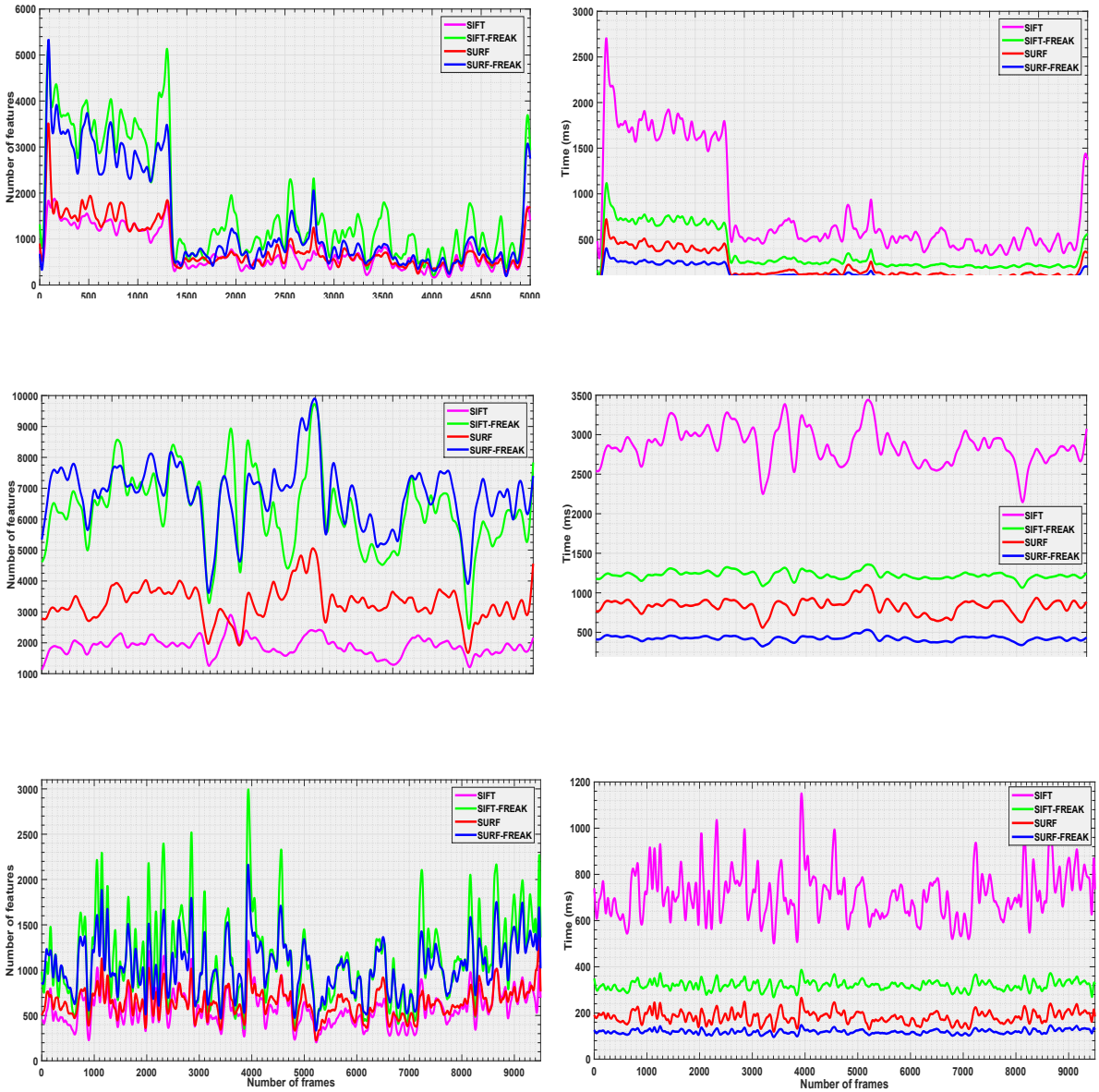
In the case of the rotation invariance property, both classical algorithms (SIFT and SURF) have a significant drop in the number of detected feature points. From Table 3.2 and Figures 3.9 and 3.10, it has been observed that the classical SIFT has accuracy less than 66%, where the classical SURF has accuracy less than 38%.

Table 3.2 Rotation Effect

	Dataset		SIFT	SURF	SIFT-FREAK	SURF-FREAK
Number of features	**G1**	15°	735↓	833↓	1682	1337↑
		30°	796↓	754↓	1679	1282↑
	G2	15°	1890↓	3293↓	6267	6793↑
		30°	1933↓	3372↓	6017	5973↑
	G3	15°	579↓	645↓	1128	1041↑
		30°	619↓	578↓	1141	1046↑
Time (ms)	**G1**	15°	757.5	790.8↑	319.7	118.4
		30°	1048.9↑	278.4↑	345.5	175.2
	G2	15°	2908.7↓	846.6	1067	359.2
		30°	3626.3↑	1186.7↑	1127.8	580.4
	G3	15°	599.3	168.1	280.8↓	125.3
		30°	614.5	180.4	282.5↓	112.7

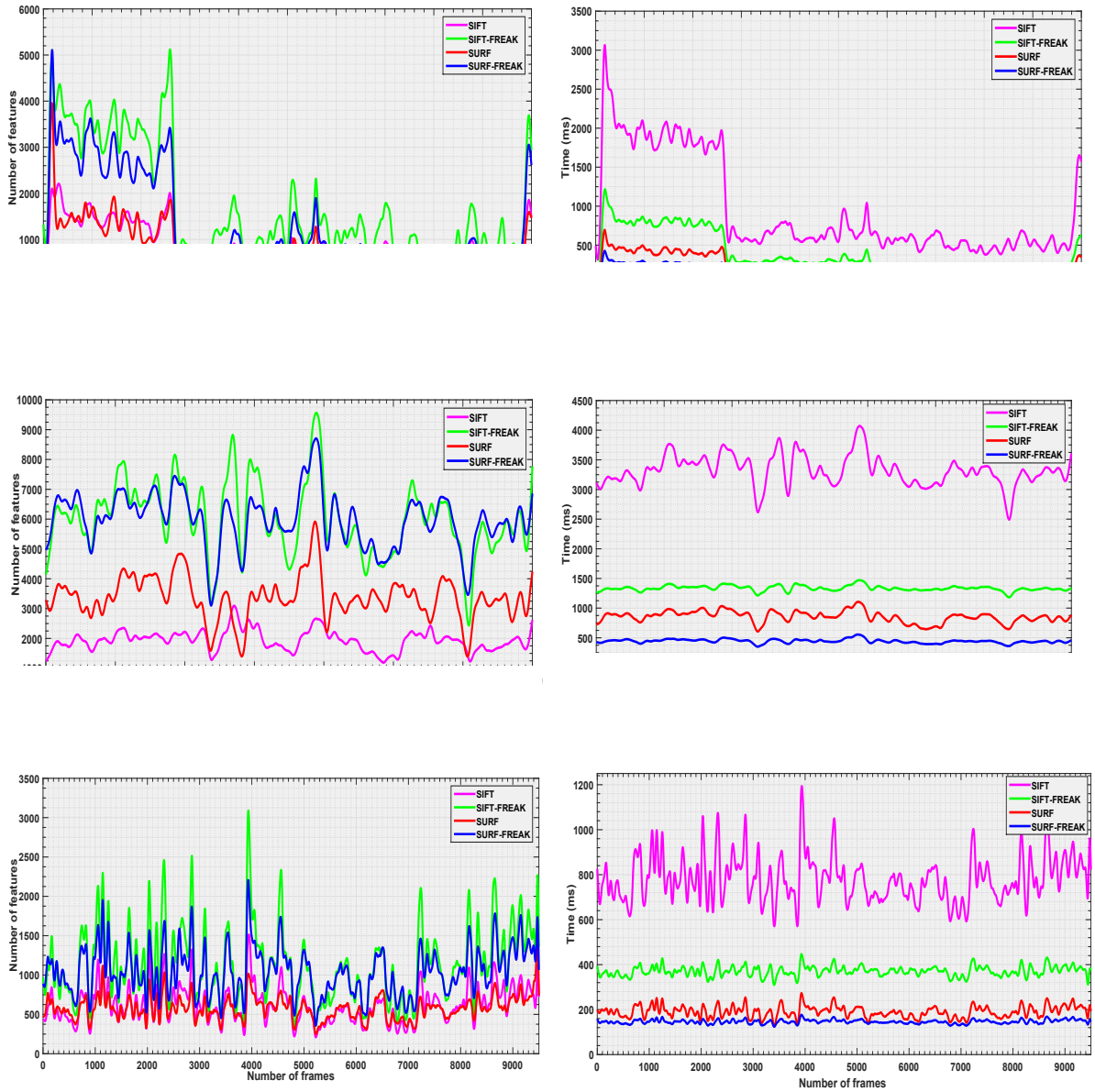
↑ Higher value than the reference. ↓ Lower value than the reference.

Moreover, by studying the effect of FREAK descriptor with SIFT and SURF detectors, it has been found that the number of the matched feature points extracted by SIFT-FREAK is almost equal to the number of feature points extracted from reference images with accuracy more than 91%. Whilst combining FREAK with SURF reduces the accuracy to less than 11%; and this is due to the detection of a large



(e) (f)

Figure 3.9 Rotated images **(15^0); a, c, e:** feature points and **b, d, f:** Computational time of 1^{st}, 2^{nd}, 3^{rd} dataset respectively.



(e) (f)

Figure 3.10 Rotated images **(30^o); a, c, e:** feature points and **b, d, f:** Computational time of 1st, 2nd, 3rd dataset respectively.

Whereas, by comparing the methods in terms of computation time, the SIFT is considered as the slowest method with computation time almost equal to the time of reference images (less number of feature and expensive computation time). Similarly to SIFT, the computational time of the SURF method is equal to the reference, and in some cases, it has a computation time higher than in the reference. While in the case of SIFT-FREAK combination, there is almost no change in the time, which

can be observed that the SIFT-FREAK combination is invariant in rotation effect with no change in computation time.

Noise invariance

For the noise effect, the classical methods still have lower accuracy in the detection of the feature points, which is less than 60% in the case of SIFT algorithm, and less than 14% for the SURF, as it illustrated in Table 3.3. Whilst, SIFT-FREAK combination provides accuracy of [91% - 94%] according to the type of the noise.

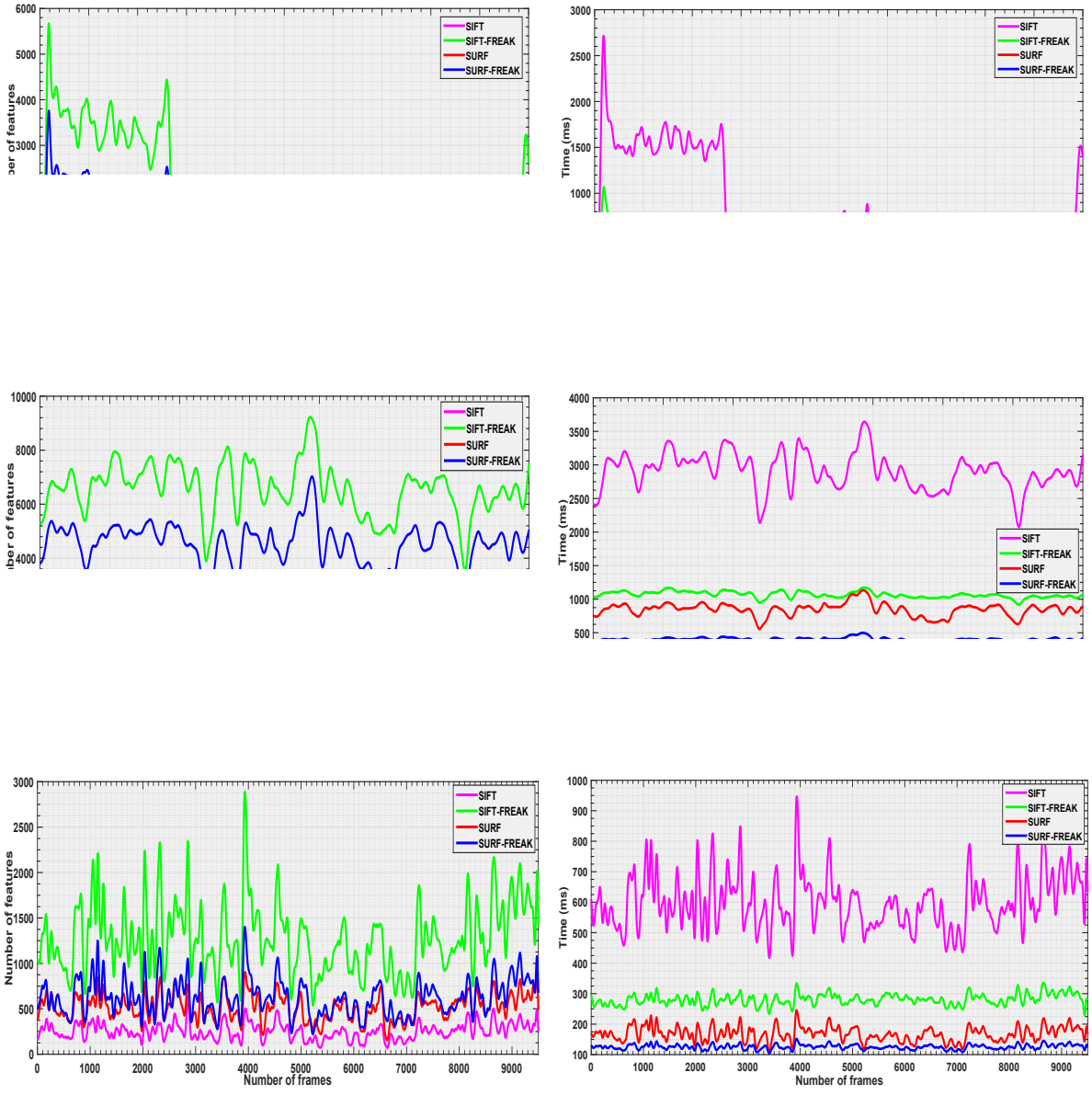
Similarly to the rotation effect, the SURF-FREAK pair suffers from detecting a large number of false positives, which in turn, reduces the accuracy significantly to 11%.

Table 3.3 Noise Effect

	Dataset		SIFT	SURF	SIFT-FREAK	SURF-FREAK
Number of features	G1	Gaus.	251↓	519↓	1631	868
		S&P	481↓	1000	1043↓	884
		S	247↓	463↓	1358↓	627↓
	G2	Gaus.	577↓	1433↓	6563↑	1067↑
		S&P	119↓	2919↓	6371↑	4731↑
		S	568↓	1272↓	6832↑	4820↑
	G3	Gaus.	253↓	551↓	1276↑	627
		S&P	103↓	300↓	1327↑	573
		S	243↓	447↓	1142	515↓
Time (ms)	G1	Gaus.	757.5	790.8↑	319.7↓	118.4
		S&P	1048.9↑	278.4↑	345.5↓	175.2↑
		S	766.9	180.5	319.0↓	121.2
	G2	Gaus.	2908.7↓	846.6	1067	395.2
		S&P	3626.4↑	1186.7↑	1127.8	580.4↑
		S	2744.8↑	880.7	1029.7	421.1
	G3	Gaus.	599.3	168.1↓	280.8↓	125.3
		S&P	614.5	180.4	282.5↓	112.7↑
		S	604.9	171.0↓	275.9↓	105.1↓

↑ Higher value than the reference. ↓ Lower value than the reference.

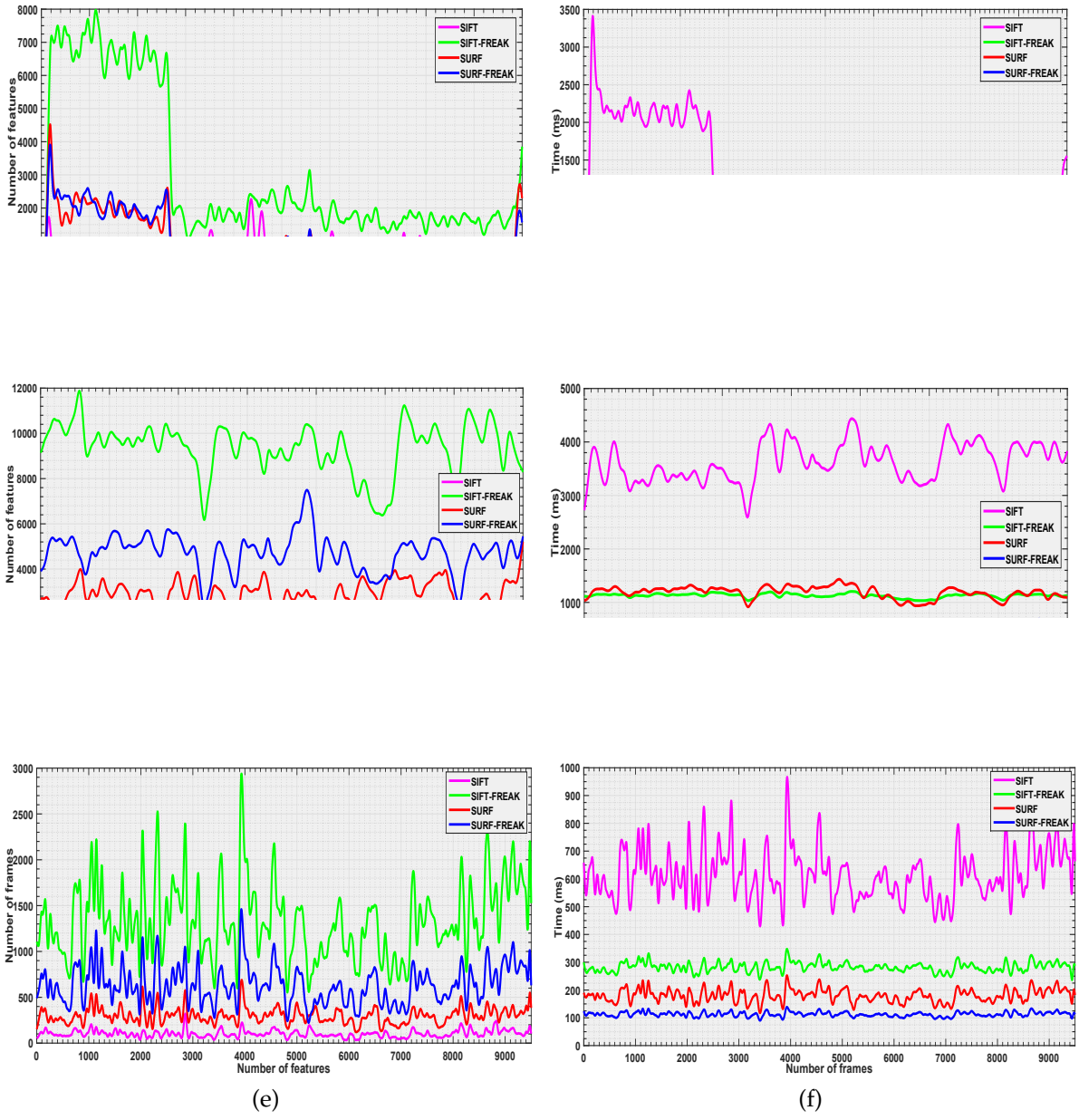
On the other hand, comparing to the reference data, the computational time of the classical SIFT in the first and second dataset, has increased significantly in the case of the *Salt-and-Pepper* noise, but it provided the same computational time with the *Gaussian* and *Speckle* noises. While generally, the other methods obtained the



(e) (f)

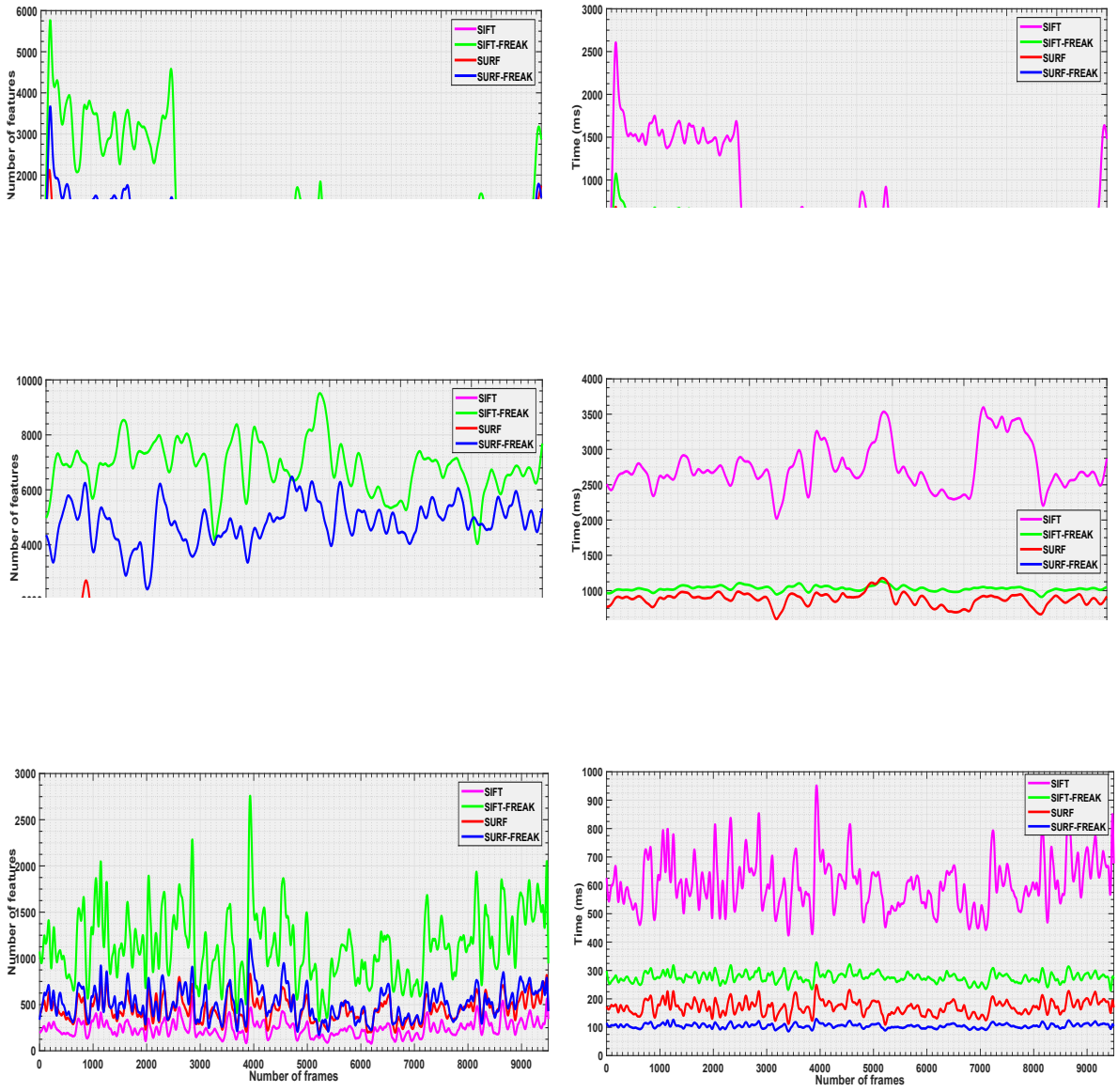
Figure 3.11 Noised images **(Gaussian)**; **a, c, e:** feature points and **b, d, f:** Computational time of 1^{st}, 2^{nd}, 3^{rd} dataset respectively.

However, SURF provided higher computation time in some cases; such as with the effect of the *Gaussian* noise in the the first dataset, and *Salt-and-Pepper* in the second dataset. Furthermore, SIFT-FREAK provided less computation time in the



(e) (f)

Figure 3.12 Noised images **(Salt-and-Pepper)**; **a, c, e:** feature points and **b, d, f:** Computational time of 1^{st}, 2^{nd}, 3^{rd} dataset respectively.

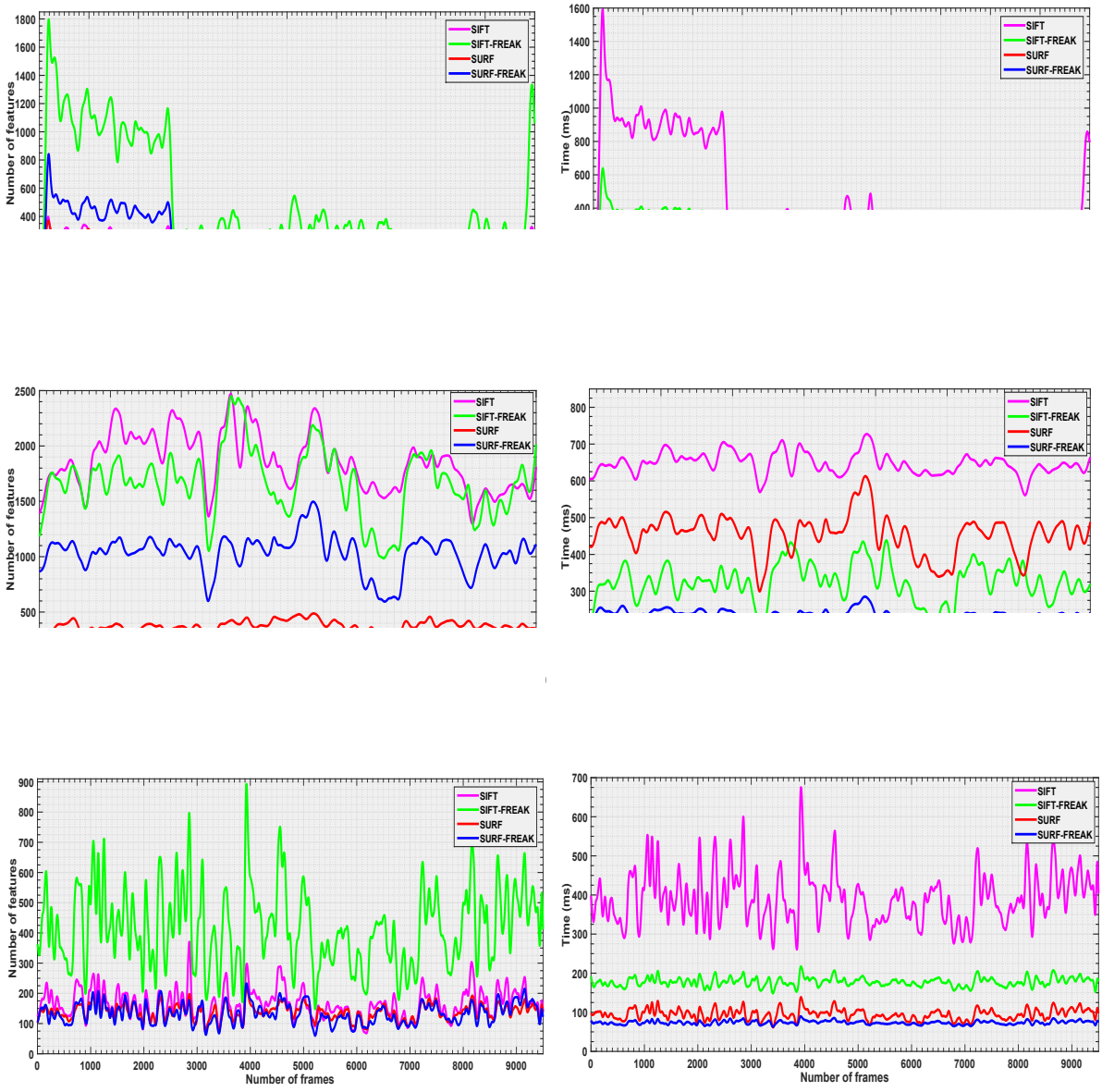


(e) (f)

Figure 3.13 Noised images **(Speckle)**; **a, c, e:** feature points and **b, d, f:** Computational time of 1^{st}, 2^{nd}, 3^{rd} dataset respectively.

Scale invariance

For the scale invariance property, Table 3.4 and Figure 3.14 show that the four algorithms provide very low accuracy of detecting the feature points, however, SIFT-FREAK combination obtained the best results with accuracy 30% compared to the classical SURF (16%), and classical SIFT (11%), while the SURF-FREAK obtained the lowest accuracy (7.6%).



(e) (f)

Figure 3.14 Scaled images; **a, c, e:** feature points and **b, d, f:** Computational time of 1^{st}, 2^{nd}, 3^{rd} dataset respectively.

From the experiments, as it is shown in Figure 3.15, the SIFT-FREAK combination provides total accuracy up to 97.7% , followed by the classical SIFT provides accuracy up to 66.8%, whilst the accuracy of the SURF algorithm is up to 38.9%, and finally the SURF-FREAK has the lowest accuracy, which is less than 7.9%.

Table 3.4 Scale Effect

	Dataset	SIFT	SURF	SIFT-FREAK	SURF-FREAK
Number of features	**G1**	164↓	131↓	484↓	171↓
	G2	1854↓	375↓	1662↓	1033↓
	G3	165↓	139↓	418↓	133↓
Time (ms)	**G1**	447.7↓	107.5↓	199.9↓	83.9↓
	G2	324.9↓	455.4↓	147.1↓	232.7↓
	G3	391.9↓	95.0↓	177.7↓	73.7↓

↑ Higher value than the reference. ↓ Lower value than the reference.

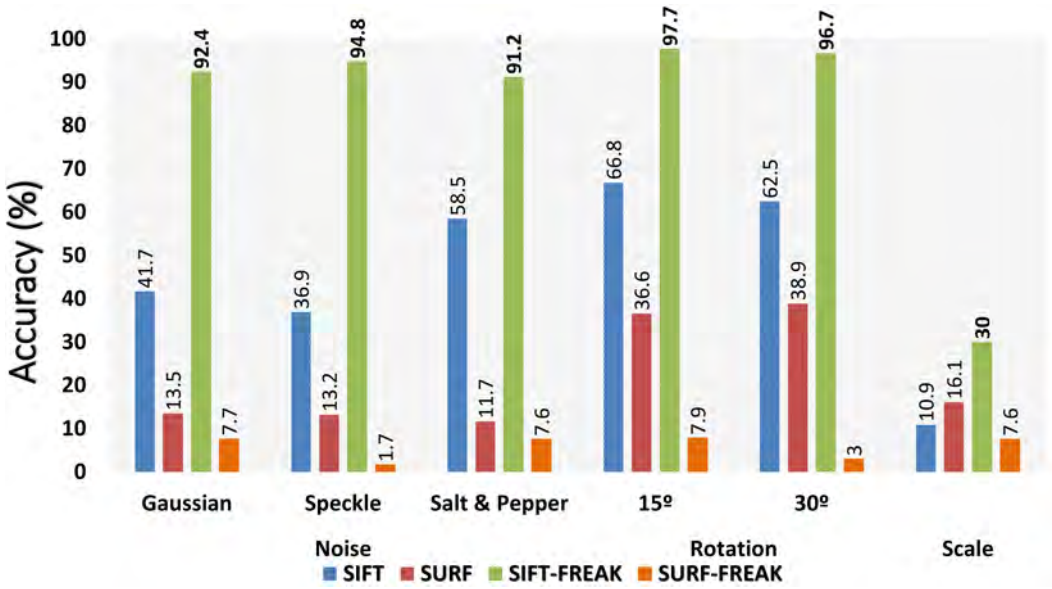


Figure 3.15 Accuracy of feature detection and description algorithm

Moreover, the obtained results from the three datasets show that the SIFT-FREAK has an average in the deviation of ± 0.2, which means that it is invariance against image transformations (*Noise*, *Rotation* and *Scale*), while SURF-FREAK is very sensitive of any small change in the image with a deviation of ± 3.4, as it is illustrated in Figure 3.16.

In addition, Table 3.5 shows the computation time required per feature point, at which, it is illustrated that the SIFT has an average time of $1.76ms$, and SURF time is $0.49ms$. Whilst the effect of the FREAK descriptor minimizes the time of processing, which has been shown that SIFT-FREAK combination has an average time $0.23ms$,

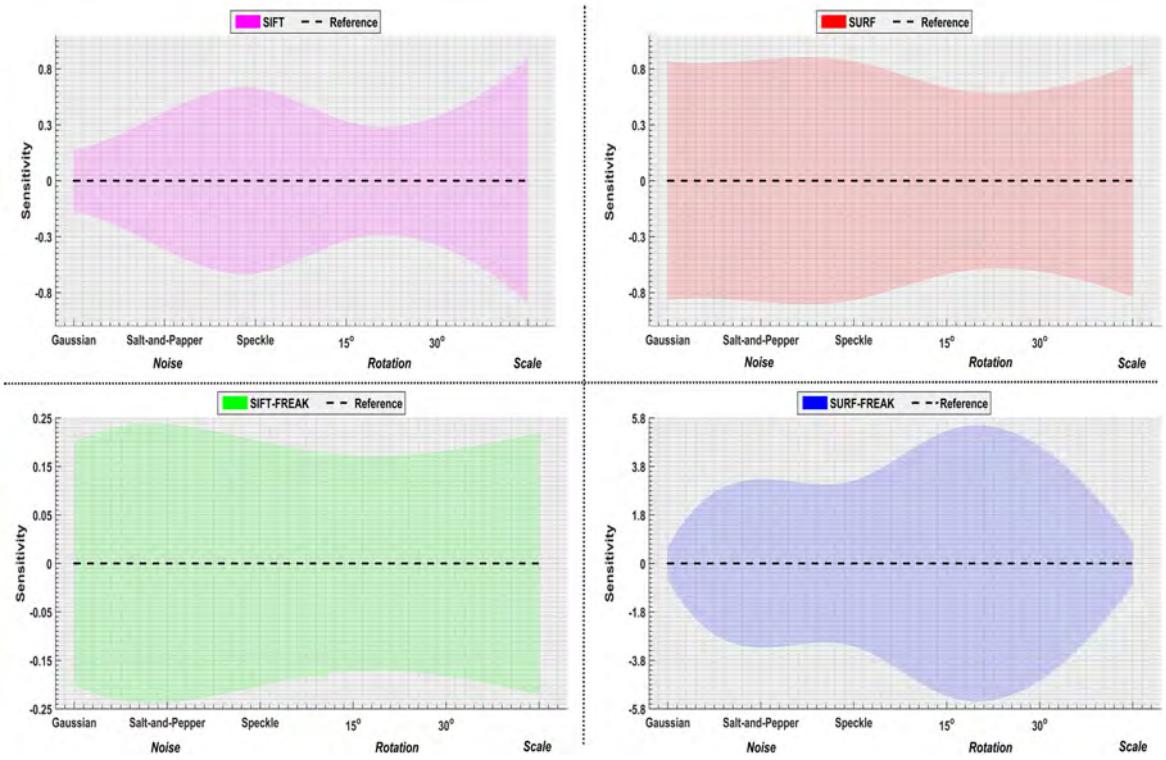


Figure 3.16 Deviation of each algorithm against image transformation

and the SURF-FREAK time is 0.23ms. Moreover, Table 3.6 shows the comparison of the average time per frame among the SIFT, SURF, SIFT-FREAK and SURF-FREAK for the three datasets.

Table 3.5 Average computation time required per feature point

			SIFT	SIFT-FREAK	SURF	SURF-FREAK
Time (ms)	**Reference**		0.441	0.257	0.171	0.444
	Noise	**Gaus**	3.476	0.193	0.835	0.236
		S&P	12.874	0.240	0.429	0.173
		Spckle	3.477	0.209	0.488	0.162
	Rotation	**15°**	1.202	0.203	0.489	0.0809
		30°	1.396	0.214	0.344	0.114
	Scale		1.760	0.309	0.906	0.424

Table 3.6 Average time required to match one image

	SIFT	SIFT-FREAK	SURF	SURF-FREAK
Time (ms)	980.9	390.12	522.737	157.38

The second evaluation step it to verified the data gathered by each algorithm (detector and descriptor) in the pose estimation algorithm, and compare the results with the ground truth and the data obtained by the DGPS. In order to achieve this, the UAV performs a trajectory of an outdoor flight (2^{nd} dataset). In in Figure 3.17, it is shown that on one hand, the SIFT-FREAK pair estimates the pose of the UAV accurately compared to the ground truth and the DGPS, followed by the SIFT algorithm. This illustrates the efficiency of the SIFT detector, However, combining the FREAK descriptor increases the robustness of the feature point extraction step. On the other hand, both SURF and SURF-FREAK algorithms provide inaccurate results with high error rates, and this is because of the large number of false positives detected.

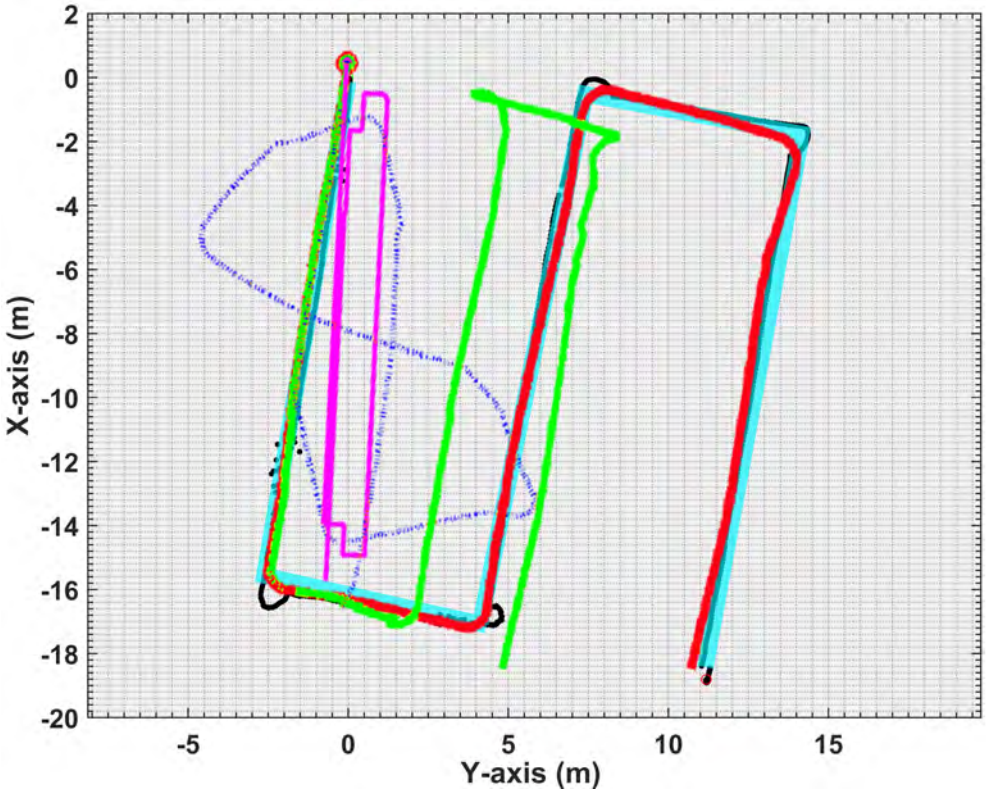


Figure 3.17 Pose estimation of outdoor flight (2^{nd} dataset) (cyan: Ground truth, red: SIFT-FREAK, green: SIFT, magenta: SURF, blue: SURF-FREAK and black: DGPS)

3.4 Homography formulation

According to Algorithm 1, first, the keypoints are detected, and its descriptors are extracted using SIFT-FREAK combination from the two consecutive frames, that are captured during the UAV motion.

Afterwards, Brute-Force algorithm is applied in order to match the keypoints from the two frames, and only the points (found in both frames) are returned. Figure 3.18 illustrates an example of matched points between two consecutive frames.

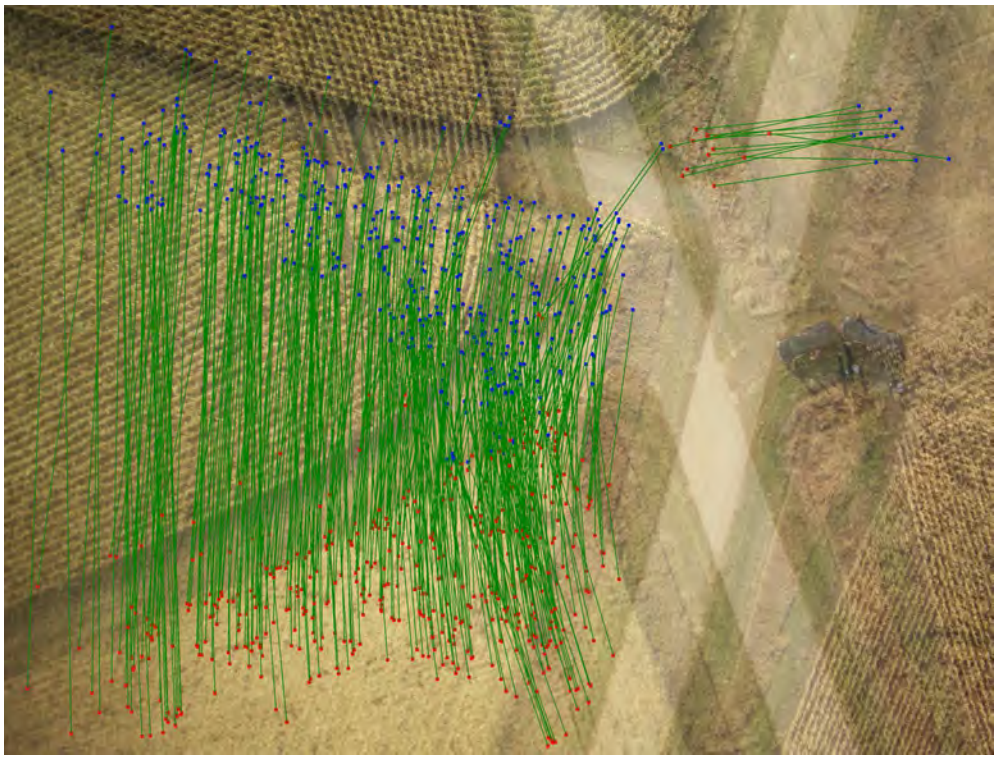


Figure 3.18 SIFT-FREAK-based Feature Matching: red points detected from F_i and blue points detected from F_{i-1}

The matching process is achieved by identifying the nearest neighbors of the detected keypoints. Therefore, for more accuracy, the matched keypoints are filtered by eliminating the ones whose ratio of the closest-distance to second-closest distance is greater than an empirical threshold value (0.28).

Let q is the filtered-matched keypoint which is calculated as shown in the fol-

lowing equation:

$$q(n) = \begin{cases} (x, y, s), & \text{distance ratio} \leq 0.28 \\ 0, & otherwise \end{cases} \qquad \forall n \in N \tag{3.5}$$

where, N is the total number of matched keypoints, s is the size of the keypoint (diameter), and $distanceratio$ is the minimum distance ratio of the matched keypoints.

However, the extracted keypoints usually contain a number of outliers that can be generated from poor images, blur, moving objects, changes in illumination, or other errors. These outliers can cause a significant error in the motion estimation, therefore, a RANSAC estimation algorithm [101] is used to generate candidate solutions. Using the smallest set of observations, the algorithm estimates the underlying parameters, and removes the effects of these outliers as it is shown is Algorithm 2 and Figure 3.19.

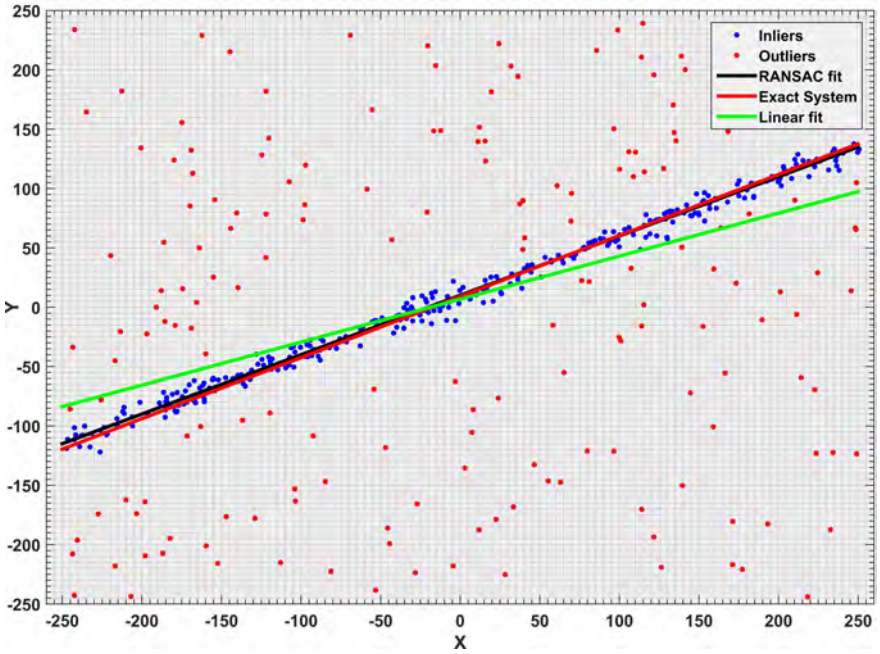


Figure 3.19 RANSAC inliers

From Algorithm 2, the error of two matched points is calculated by using the symmetric transfer error from the homography H of these points as follows:

$$d^2_{transfer} = d(q_i, H^{-1}q_{i-1})^2 + d(q_{i-1}, Hq_i)^2 \tag{3.6}$$

where, d denotes the distance and q_i and q_{i-1} are the matched points.

Algorithm 2: Inliers estimation using RANSAC

Input: Matched points q

Output: Inliers $Inlr$

Define: Maximum number of iterations N, Minimum number of points to fit the model k, Best model $bstmdl$, Best score $bstscr$, Threshold ε, Temporary inliers $tempinlr$, Temporary model $tempmdl$, Temporary score $tempscr$, Error err

begin
  $bstscr \leftarrow \infty$
  while $< N$ **do**
    $tempinlr \leftarrow$ Random(q, k)
    if *tempinlr is colinear* **then**
      $tempmdl \leftarrow$ EstimateModel$(tempinlr)$
      $tempscr \leftarrow 0$
      for $i \leftarrow 0, size(q)$ **do**
        $err \leftarrow$ distance$(tempmdl, q[i])$
        if $err < \varepsilon$ **then**
          $tempscr \leftarrow tempscr + err$
          $tempinlr \leftarrow$ Add$(q[i])$
        end
        else
          $tempscr \leftarrow tempscr + \varepsilon$
        end
      end
      if $err < \varepsilon$ **then**
        $bstscr \leftarrow tempscr$
        $bstmdl \leftarrow tempmdl$
        $Inlr \leftarrow tempinlr$
      end
    end
  end
end

Assuming set of points $q_{i-1} \in \mathbb{R}^2$ from the frame F_{i-1} related to the corresponding points $q_i \in \mathbb{R}^2$ from F_i, then the *frame-to-frame* motion in the image plane can be

extracted from the homography as defined in Equation 3.7:

$$
\begin{gathered}
sq_i = {}^{i}\mathrm{H}_{i-1} q_{i-1} \\
s \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_{i-1} \\ y_{i-1} \\ 1 \end{bmatrix}
\end{gathered} \tag{3.7}
$$

where $q_i = (x_i, y_i, 1)^T$ and $q_{i-1} = (x_{i-1}, y_{i-1}, 1)^T$ are homogeneous, s is an arbitrary scale factor, ${}^{i}\mathrm{H}_{i-1}$ is the 3×3 planner homography matrix as shown in Figure 3.20.

From Equation 3.7, two linear system equations of the elements of ${}^{i}\mathrm{H}_{i-1}$ can be obtained:

$$
\begin{aligned}
x_i \left(h_{31} x_{i-1} + h_{32} y_{i-1} + h_{33}\right) &= h_{11} x_{i-1} + h_{12} y_{i-1} + h_{13} \\
y_i \left(h_{31} x_{i-1} + h_{32} y_{i-1} + h_{33}\right) &= h_{21} x_{i-1} + h_{22} y_{i-1} + h_{23}
\end{aligned} \tag{3.8}
$$

Thees are inhomogeneous equations involving the positions of the points non-linearly, however, the homography elements are linear. Thus, Equation 3.8 can be derived as follows:

$$
\begin{aligned}
a_x^T h &= 0 \\
a_y^T h &= 0
\end{aligned} \tag{3.9}
$$

where,

$$
\begin{aligned}
a_x^T &= (x_{i-1}, y_{i-1}, 1, 0, 0, 0, -x_{i-1}x_i, -y_{i-1}x_i, -x_i) \\
a_y^T &= (0, 0, 0, x_{i-1}, y_{i-1}, 1, -x_{i-1}y_i, -y_{i-1}y_i, -y_i) \\
h &= (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33})^T
\end{aligned} \tag{3.10}
$$

Then for n points, the inhomogeneous equation can be represented by:

$$
Ah = 0, \qquad h \neq 0 \tag{3.11}
$$

where H contains the homography elements h_{ij} (ith row, jth column), and A is a

$2n \times 9$ matrix which is defined as follows:

$$A = \begin{bmatrix} x_{i-1_1} & y_{i-1_1} & 1 & 0 & 0 & 0 & -x_{i-1_1}x_{i_1} & -y_{i-1_1}x_{i_1} & -x_{i_1} \\ 0 & 0 & 0 & x_{i-1_1} & y_{i-1_1} & 1 & -x_{i-1_1}y_{i_1} & -y_{i-1_1}y_{i_1} & -y_{i_1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{i-1_n} & y_{i-1_n} & 1 & 0 & 0 & 0 & -x_{i-1_n}x_{i_n} & -y_{i-1_n}x_{i_n} & -x_{i_n} \\ 0 & 0 & 0 & x_{i-1_n} & y_{i-1_n} & 1 & -x_{i-1_n}y_{i_n} & -y_{i-1_n}y_{i_n} & -y_{i_n} \end{bmatrix} \tag{3.12}$$

It is noticed that the H matrix may fall in the null space of A, therefore, the Singular Value Decomposition (SVD) method is used in order to solve the null space issue, where:

$$A = U\Sigma V^T = \sum_{i=1}^{9} \sigma_i u_i v_i^T \tag{3.13}$$

The SVD is computed where the singular values of the diagonal of Σ are sorted descendingly Furthermore, the eigenvalues of $A^T A$ are equal to the squares of the singular values of H, so that, the value of $\sigma_9 = 0$, if H is exactly determined, which means that the homography fits the points correspondences exactly. In the case if H is overdetermined, the solution is equal to the last column in V which corresponds to the smallest singular value σ_9.

In order to estimate the 3D position in the world plane, the relation between the points in the real-world and their correspondings in the image plane should be estimated. Let a world point $q_w \in \mathbb{R}^3 \wedge q_w = (x_w, y_w, z_w, 1)^T$, then by using 3×4 projection matrix P^i, this point can be mapped to an image point $q_i \in \mathbb{R}^2$ as follows:

$$sq_i = P^i q_w = C \left[R^i | t^i \right] q_w = C \begin{bmatrix} r_1^i & r_2^i & r_3^i & t^i \end{bmatrix} q_w \tag{3.14}$$

where $C = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$ is the calibration matrix of the camera (Appendix B shows the process of calibrating the cameras), R^i and t^i are the rotation matrix and translation vector respectively.

Suppose that q_w lies on a plane $\mathscr{P}$ with $z_w = 0$, then Equation 3.14 can be re-

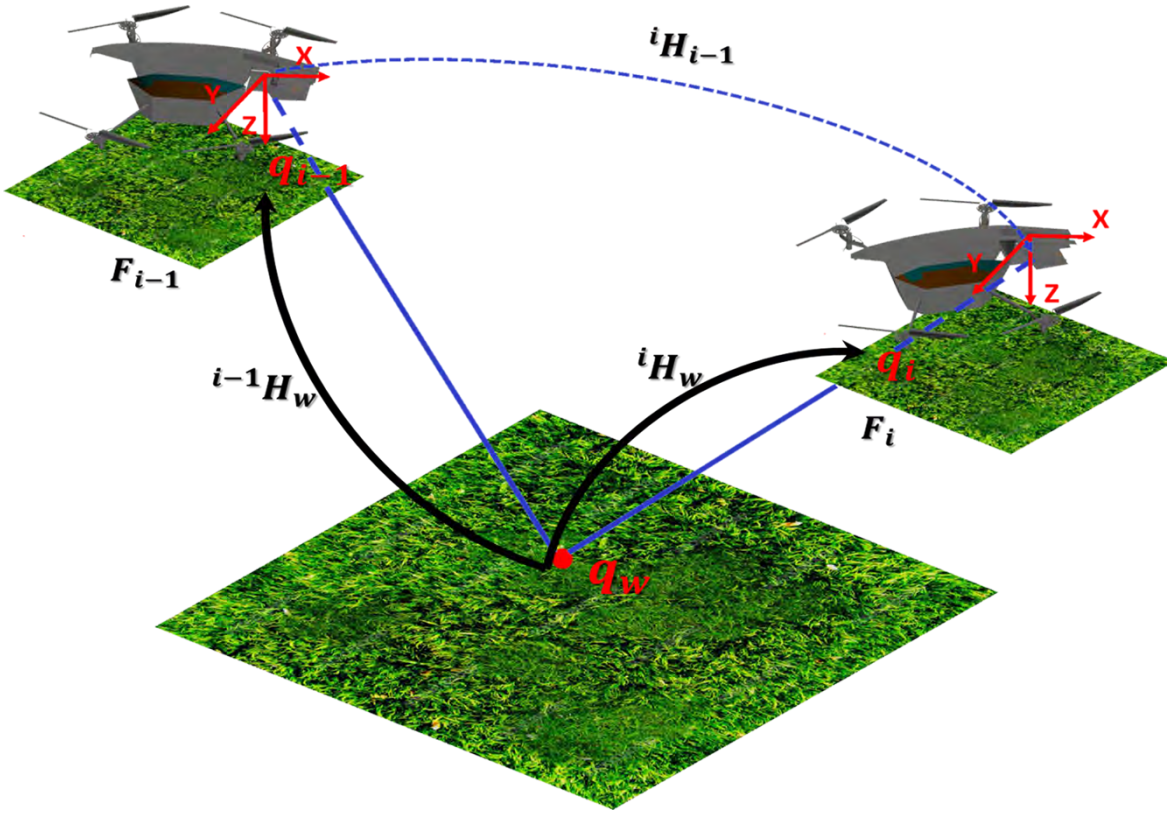


Figure 3.20 The relation between the matched point q from two consecutive frames.

expressed as follows:

$$sq_i = \acute{P}^i q_{\mathscr{P}} = C\left[R^i|t^i\right] q_{\mathscr{P}} = C\begin{bmatrix} r_1^i & r_2^i & t^i \end{bmatrix} q_{\mathscr{P}} \tag{3.15}$$

where $\acute{P}^i$ is a 3×3 projection matrix, which maps the points in the plane $\mathscr{P}$ to the image plane and can be defined as a planar homography as follows:

$${}^{i}\mathrm{H}_w = \lambda C\begin{bmatrix} r_1^i & r_2^i & t^i \end{bmatrix} = \acute{P}^i \tag{3.16}$$

where $\lambda = \frac{1}{s}$ is the homogeneous scale factor.

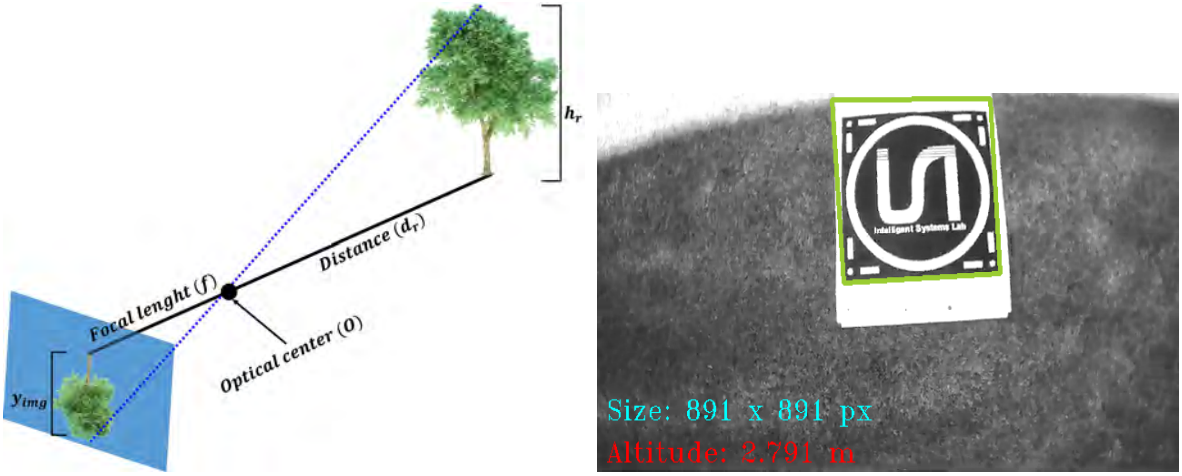
However, ${}^{i}\mathrm{H}_w$ cannot be calculated directly, therefore, it is required to calculate the initial homography ${}^{0}\mathrm{H}_w$ of the reference frame (*world-to-reference frame*). ${}^{0}\mathrm{H}_w$ is calculated as shown in Equation 3.17.

$${}^{0}\mathrm{H}_w = \lambda C\begin{bmatrix} {}^{0}\mathrm{r}_1 & {}^{0}\mathrm{r}_2 & t^0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \zeta \end{bmatrix} \tag{3.17}$$

where, ζ is the initial altitude obtained by any sensor; such as ultrasonic, barometer, etc. or it can be estimated form the size of a detected predefined pattern in the initial position of the navigation using the pinhole model as it is shown in Figure 3.21, and calculated by Equation 3.18 as follows:

$$\begin{aligned} \frac{-y_{img}}{f} &= \frac{h_r}{d_r} \quad or, \\ d_r &= -\frac{f h_r}{y_{img}} \end{aligned} \tag{3.18}$$

where y_{img} is the object hight in the image plane, h_r is the object real hight, f is the focal length of the camera, and d_r is the distance between the camera optical center and the object.



(a) (b)

Figure 3.21 Altitude Estimation from detected pattern - **a:** Pinhole Model, **b:** Altitude estimation.

Thereafter, the related i^{th} frame homographies can be estimated as a sequence of *frame-to-frame* homographies multiplication as follows:

$${}^{i}\mathrm{H}_{w} = {}^{i}\mathrm{H}_{i-1}\,{}^{i-1}\mathrm{H}_{i-2}\,\ldots\,{}^{1}\mathrm{H}_{0}\,{}^{0}\mathrm{H}_{w} \tag{3.19}$$

From equations 3.15 and 3.16, it is possible to decompose ${}^{i}\mathrm{H}_{w}$ to its rotation and

translation matrices as follows:

$$\begin{bmatrix} {}^{i}\mathrm{r}_1 & {}^{i}\mathrm{r}_2 & {}^{i}\mathrm{r}_3 & t^T \end{bmatrix} = \lambda C^{-1} {}^{i}\mathrm{H}_w \tag{3.20}$$

where,

$$\begin{aligned} r_1 &= \lambda C^{-1} h_1 \\ r_2 &= \lambda C^{-1} h_2 \\ r_3 &= r_1 \times r_2 \\ t &= \lambda C^{-1} h_3 \end{aligned} \tag{3.21}$$

and

$$\lambda = \frac{1}{\|C^{-1}h_1\|} = \frac{1}{\|C^{-1}h_2\|} \tag{3.22}$$

The rotation matrix R is an orthogonal matrix; hence $R^T R = RR^T = I$. However, when solving the problem using real data, it will not end up with an exact rotation matrix for which the property holds. So that, the SVD of R is used to get around this problem:

$$R = U\Sigma V^T \tag{3.23}$$

Because R is itself orthonormal, the matrix Σ must be the identity matrix, such that the new rotation matrix can be obtained as follows:

$$\acute{R} = \begin{bmatrix} \acute{r}_1 & \acute{r}_2 & \acute{r}_3 \end{bmatrix} = UIV^T \tag{3.24}$$

According to the notational conventions, and ignoring the various possible axis permutations discussed by Ken Shoemake [281], the proposed algorithm has focused on one particular order of applying rotations to illustrate the problem. Defining the three standard sequence of rotation matrices: $R_x(\phi_{roll})$ about the *x-axis*,

$R_y(\theta_{pitch})$ about the *y-axis* and $R_z(\psi_{yaw})$ about the *z-axis* as:

$$R_x(\phi_{roll}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi_{roll}) & \sin(\phi_{roll}) \\ 0 & -\sin(\phi_{roll}) & \cos(\phi_{roll}) \end{bmatrix}$$

$$R_y(\theta_{pitch}) = \begin{bmatrix} \cos(\theta_{pitch}) & 0 & -\sin(\theta_{pitch}) \\ 0 & 1 & 0 \\ \sin(\theta_{pitch}) & 0 & \cos(\theta_{pitch}) \end{bmatrix} \tag{3.25}$$

$$R_z(\psi_{yaw}) = \begin{bmatrix} \cos(\psi_{yaw}) & \sin(\psi_{yaw}) & 0 \\ -\sin(\psi_{yaw}) & \cos(\psi_{yaw}) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The final coordinate transformation matrix of the rotations $R_x(\phi_{roll})$, $R_y(\theta_{pitch})$ and $R_z(\psi_{yaw})$ can be written as:

$$R_{xyz} = \begin{bmatrix} \underline{c_\theta c_\psi} & \underline{c_\theta s_\psi} & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta \end{bmatrix} \tag{3.26}$$

where, $c_\theta = cos(\theta_{pitch})$, $s_\phi = cos(\phi_{roll})$ and so on. From Equation 3.26, the three angles are extracted as:

$$\begin{aligned} \phi_{roll} &= atan2(s_\phi c_\theta, c_\phi c_\theta) \\ \theta_{pitch} &= atan2(-s_\theta, c_\phi) \\ \psi_{yaw} &= atan2(c_\theta s_\psi, c_\theta c_\psi) \end{aligned} \tag{3.27}$$

where $c_\phi = \sqrt{(c_\theta c_\psi)^2 + (c_\theta s_\psi)^2}$.

A problem appears when the value of $c_\theta s_\psi$ and $c_\theta c_\psi$ is very small or equal to zero, which yields $s_\phi c_\theta$ and $c_\phi c_\theta$ to be very small or equal to zero, and $-s_\theta$ will be in the range close to ± 1. At which, it is difficult to estimate the angle ϕ_{roll}. However, this is solved by comparing the value of c_θ with an empiric threshold value ($\varepsilon = 1.9073e^{-6}$), and if c_θ falls below this threshold, the matrix elements are

reduced to the following:

$$R_{xyz} = \begin{bmatrix} 0 & 0 & -(\pm 1) \\ \pm s_\phi c_\psi - c_\phi s_\psi & \pm s_\phi s_\psi + c_\phi c_\psi & 0 \\ \pm c_\phi c_\psi + s_\phi s_\psi & \pm c_\phi s_\psi - s_\phi c_\psi & 0 \end{bmatrix} \tag{3.28}$$

This matrix leads to an example of a gimbal lock, at which the x and z axes are brought into alignment by the y rotation, and loses one degree of freedom because ϕ_{roll} and ψ_{yaw} angles act as a single parameter.

Handling this case by setting one of ϕ_{roll} or ψ_{yaw} to zero, allows to derive the other angle, and thus reduces the matrix to the following form:

$$R_{xyz} = \begin{bmatrix} 0 & 0 & -(\pm 1) \\ \pm s_\phi & c_\phi & 0 \\ \pm c_\phi & -s_\phi & 0 \end{bmatrix} \tag{3.29}$$

Hence, ϕ_{roll} can be easily extracted as follow:

$$\phi_{roll} = atan2(-s_\phi, c_\phi) \tag{3.30}$$

Working with real-world matrices whose elements have typical rounding errors, ϕ_{roll} sometimes can be obtained with some confused set of values, which lead to produce abrupt changes in the angle values. However, this is not a major problem, and that is because of the close relationship between ϕ_{roll} and ψ_{yaw} as mentioned before. Therefore, any abrupt change in the value of ϕ_{roll} can be easily corrected by the value of ψ_{yaw}.

On the other hand, if the value of c_θ exceeds the threshold ε, each of the angles ϕ_{roll} and ψ_{yaw} acts independently, and the rounding errors in one angle become very large and cannot be corrected by the other.

The solution can be achieved by computing the rotation generated by the extracted ϕ_{roll} and θ_{pitch}, and then working out the rotation required in ψ_{yaw} in order to reconstruct its matrix. This can be derived by multiplying the unknown ψ_{yaw}

matrix by the transpose of the product of ϕ_{roll} and θ_{pitch} matrices as follow:

$$\begin{aligned}\acute{R} &= \left[\begin{bmatrix}1 & 0 & 0\\ 0 & c_\phi & s_\phi\\ 0 & -s_\phi & c_\phi\end{bmatrix}\begin{bmatrix}c_\theta & 0 & -s_\theta\\ 0 & 1 & 0\\ s_\theta & 0 & c_\theta\end{bmatrix}\right]^T\begin{bmatrix}\acute{r}_{11} & \acute{r}_{12} & \acute{r}_{13}\\ \acute{r}_{21} & \acute{r}_{22} & \acute{r}_{23}\\ \acute{r}_{31} & \acute{r}_{32} & \acute{r}_{33}\end{bmatrix}\\ &= \begin{bmatrix}c_\theta & s_\phi s_\theta & c_\phi s_\theta\\ 0 & c_\phi & -s_\phi\\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta\end{bmatrix}\begin{bmatrix}\acute{r}_{11} & \acute{r}_{12} & \acute{r}_{13}\\ \acute{r}_{21} & \acute{r}_{22} & \acute{r}_{23}\\ \acute{r}_{31} & \acute{r}_{32} & \acute{r}_{33}\end{bmatrix}\end{aligned} \tag{3.31}$$

As the product represents a pure rotation about the *z-axis*, then ψ_{yaw} must be in the following form:

$$\acute{R} = \begin{bmatrix}c_\psi & s_\psi & 0\\ -s_\psi & c_\psi & 0\\ 0 & 0 & 1\end{bmatrix} \tag{3.32}$$

Accordingly, from equations 3.31 and 3.32, the ψ_{yaw} can be extracted as follows:

$$\psi_{yaw} = atan2(s_\psi, c_\psi) = atan2((s_\phi \acute{r}_{31} - c_\phi \acute{r}_{21}), (c_\phi \acute{r}_{22} - s_\phi \acute{r}_{32})) \tag{3.33}$$

3.5 Experimental Results

The proposed algorithm explained in the previous sections has been tested and verified with data gathered from real flights in both indoor and outdoor arenas with different visual conditions; such as ground texture, illumination, flight altitude, etc.

The processing in the ground station is performed in Intel i7-3770 at 3.4 GHz CPU. The connection with the UAV is established via a standard 802.11n wireless LAN card.

3.5.1 Platform

As mentioned before, two sets of experiments were performed (indoor and outdoor), different UAV platform was used in each test. The indoor experiments have been carried out with Parrot AR.Drone 2.0 quadcopter [49, 170], which supplies data from a three-axis gyroscope and accelerometer, pressure sensor and ultrasound sensor for ground altitude measurement. In addition, the vehicle is provided with two cameras, a 92^o diagonal angle lens frontal camera provides 1280×720

color images where the vertical camera has 64^o diagonal angle and provides color images with 320×240 resolution.

In the outdoor experiments, DJI F450 quadcopter based on Pixhawk control system was used. The quadcopter is equipped with GPS, magnetometer and IMU (accelerometers, gyroscopes, and barometer). In addition, a SJ4000 wireless camera which provides 640×480 images is mounted on Walkera G-2D gimbal to provide stability of the camera.

3.5.2 Results

In the experiments, different flights are performed in two different scenarios, in order to test and evaluate the performance of the proposed algorithm in both accuracy and processing time.

In these experiments, the frames are captured and sent to the ground station every 120 ms, then the system extracts the feature points from each new frame and match them with feature points from the previous frame. By computing and decomposing the *frame-to-frame* and *frame-to-world* homographies, the system estimates robustly the pose of the UAV in terms of positions (x, y, z) and orientations (ϕ, θ, ψ).

Outdoor

The first scenario is the outdoor flights. At which, the flights are performed following a predefined path (ground truth), with altitudes between 1 m to 4 m. The results obtained from the visual system are compared with the data of the DGPS system.

Figures 3.22a, 3.23a, and 3.24a illustrate examples of the displacements in the three axes (t_x, t_y, t_z). While, Figures 3.22b, 3.23b, and 3.24b show the Euler angles (ϕ, θ, ψ) estimated by the vision system.

In Figure 3.22c the UAV performs a rectangle-shape outdoor flight with total distance of 117.6 m. In this experiment, the UAV performs a combination of straight line motion and turn to left side; in order to test the ability of the pose estimation to return to the starting point. From the figure (detailed image) it has been observed that during the flight, in different parts of the trajectory, the DGPS loses its accuracy, while in contrast, the vision system maintains its position with respect to the ground truth. The final error generated by the vision system was 15 cm comparing to 48 cm generated by the DGPS.

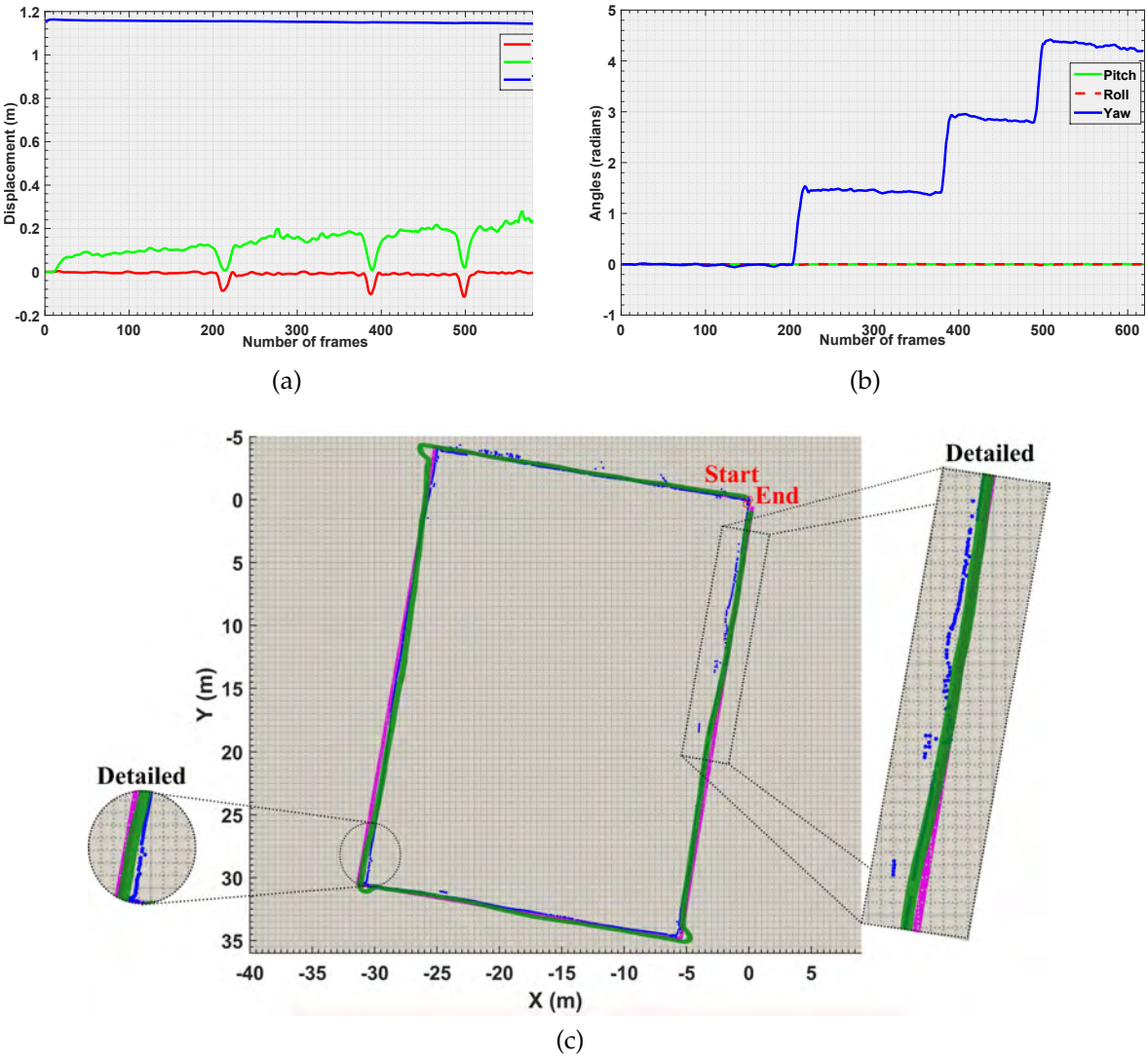


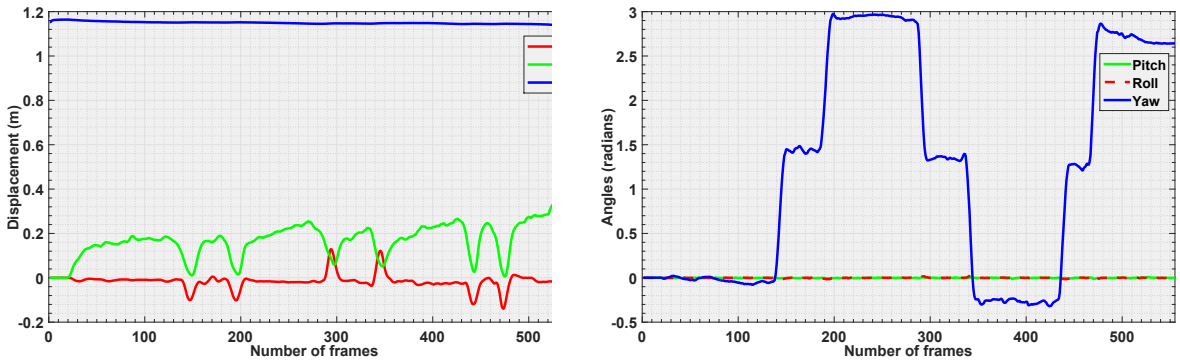
Figure 3.22 Outdoor flight - Rectangle shape - total distance: 117.6m; **a:** Translation vector (vision), **b:** Euler angles(vision), and **c:** Pose estimation in 2D (Ground truth, Vision and DGPS).

In the second outdoor experiment, the UAV flies in a square wave-shape with total distance of 124.9 *m*. As it is illustrated in Figure 3.23c, in this experiment, the UAV moves in straight lines and turning in both sides (left and right). The same problem appeared with the DGPS, which its accuracy decreased in different parts in the trajectory during the flight (detailed image). The cumulative error obtained by the vision system was less than 5 *cm* where the DGPS error was 90 *cm*.

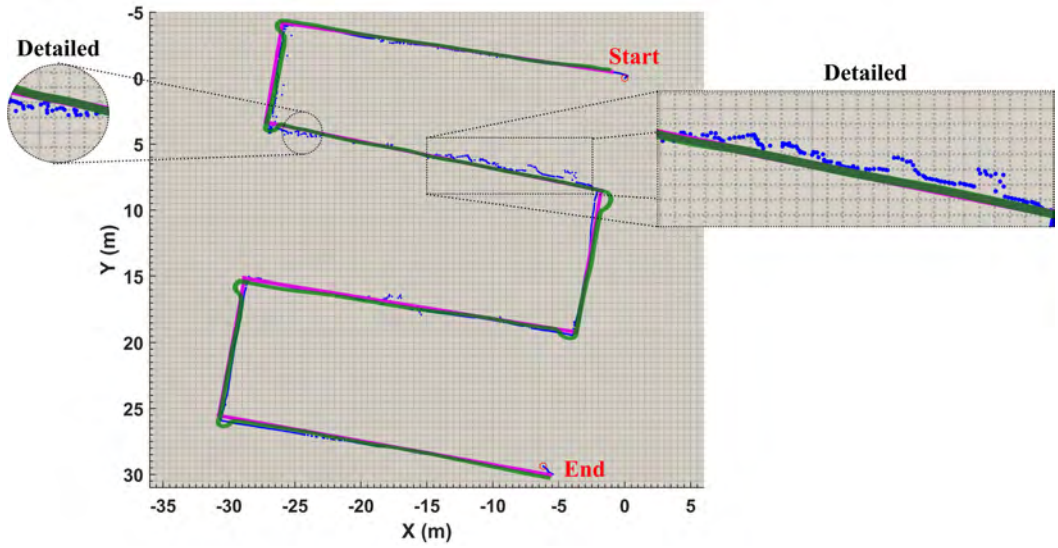
The last experiment is shown in Figure 3.24c, where the UAV performs a random-shape flight. In this experiment, it is observed that the DGPS loses its accuracy from

the beginning because the ***Start*** point is close to a building wall. In addition, it provides an imprecise signal data with error of 75 *cm* in a comparison to the ground truth. In contrast, the vision system was able to follow the path and keep the position as the ground truth. The total error obtained by the vision system is 50 *cm* while the DGPS error is 110 *cm*.

From the outdoor experiments, the maximum error generated by the vision system is $\pm$ 50 *cm* in *(x,y)-axis* and $\pm$ 30 *cm* in *z-axis*.

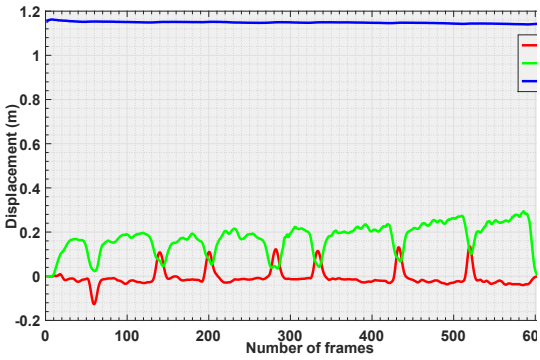


(a) (b)

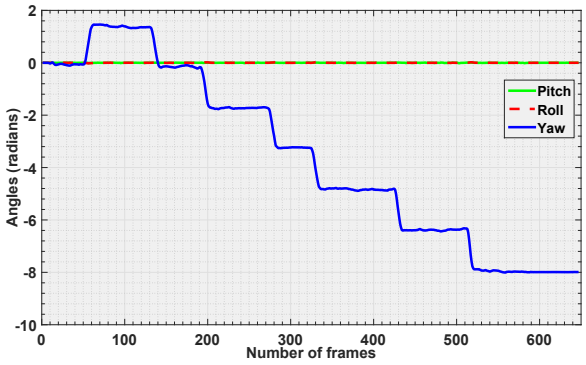


(c)

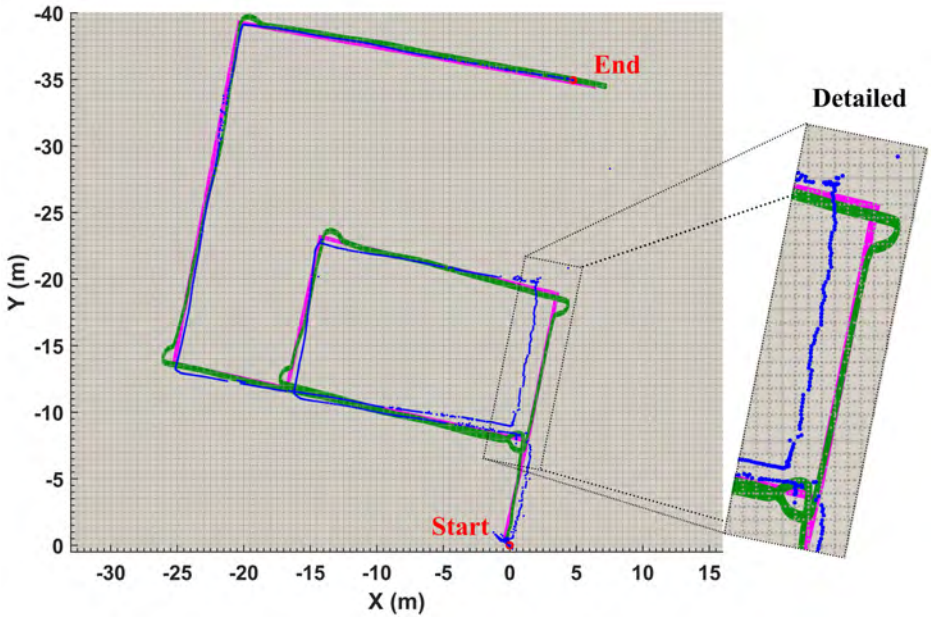
Figure 3.23 Outdoor flight - Square wave shape - total Distance: 124.9m; **a:** Translation vector (vision), **b:** Euler angles(vision), and **c:** Pose estimation in 2D (Ground truth, Vision and DGPS).



(a)



(b)



(c)

Figure 3.24 Outdoor flight - Random shape - total distance: 132.3m; **a:** Translation vector (vision), **b:** Euler angles(vision), and **c:** Pose estimation in 2D (Ground truth, Vision and DGPS).

Indoor

The second scenario is the indoor flights. At which, free flying movements were performed with a maximum altitude of 2.5 m, and the estimated results are compared with the results obtained by 9-cameras VICON motion capture system (Experiments performed in the department of Aeronautical and Automotive Engineering - Loughborough University).

Similarly to the outdoor experiments, the motion displacements in the three

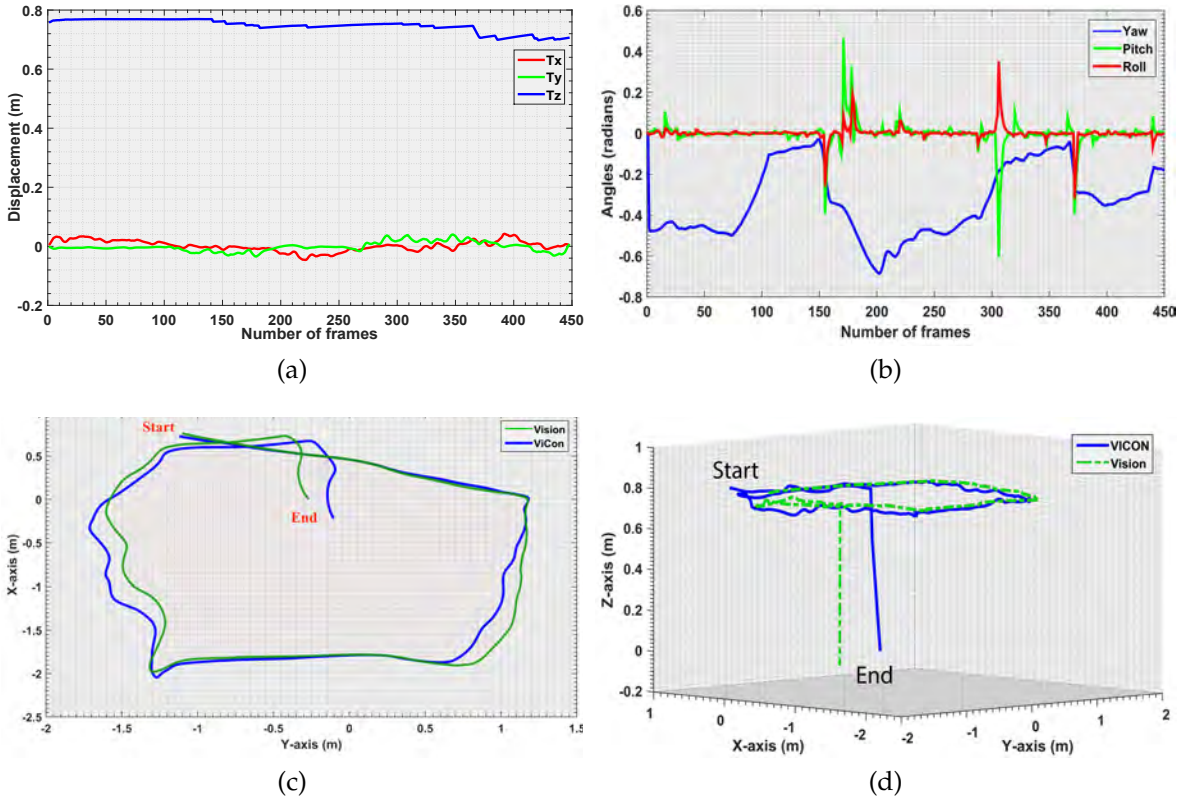


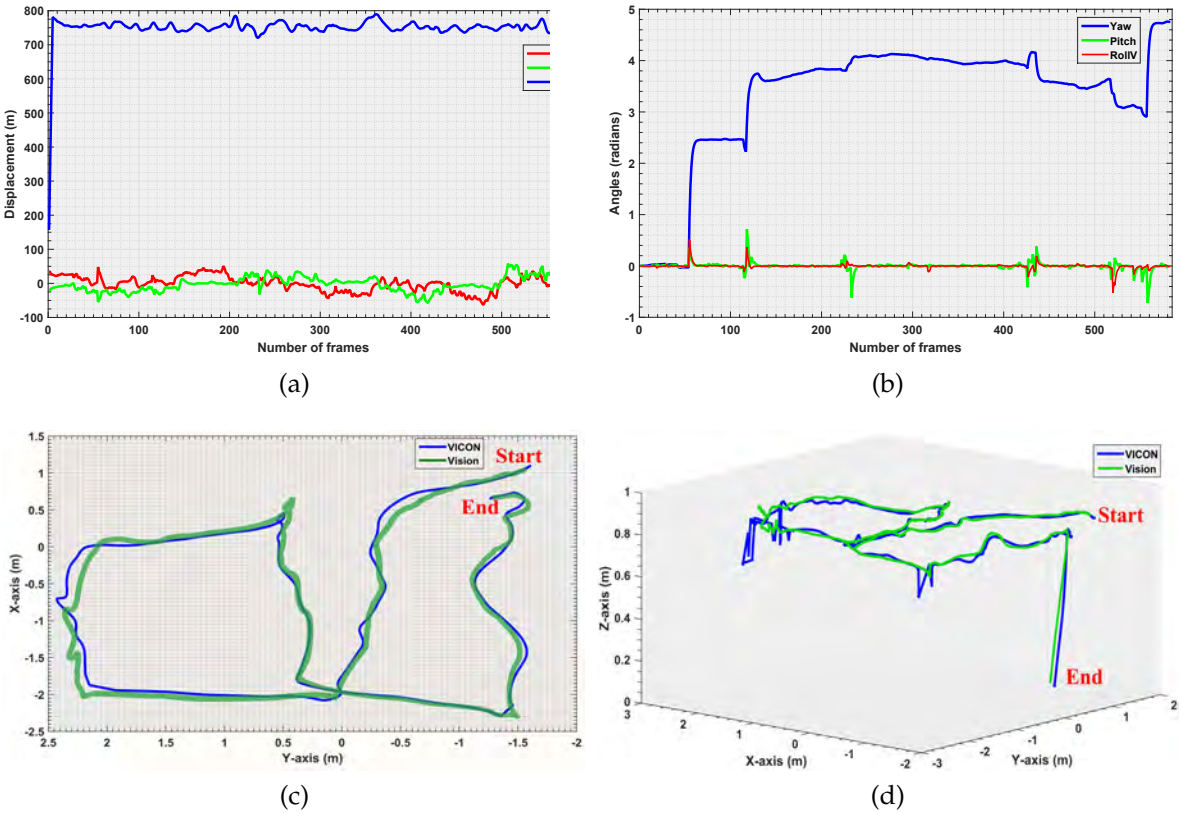
Figure 3.25 Indoor flight: **a:** Translation vector (vision), **b:** Euler angles(vision), **c:** Pose estimation in 2D, and d) Pose estimation in 3D.

axes (t_x, t_y, t_z) are shown in Figures 3.25a and 3.26a, while, Figures 3.25b and 3.26b show the Euler angles (ϕ, θ, ψ) estimated by the vision system.

Figures 3.25c and 3.26c show the UAV the estimated 2D motion by the VICON compared to the Vision system. Whilst Figures 3.25d and 3.26d show the 3D motion obtained by the two systems. From these experiments, it was found that the maximum error generated by the vision system is $\pm$ 35 *cm* in *(x,y)-axis* and $\pm$ 20 *cm* in *z-axis*.

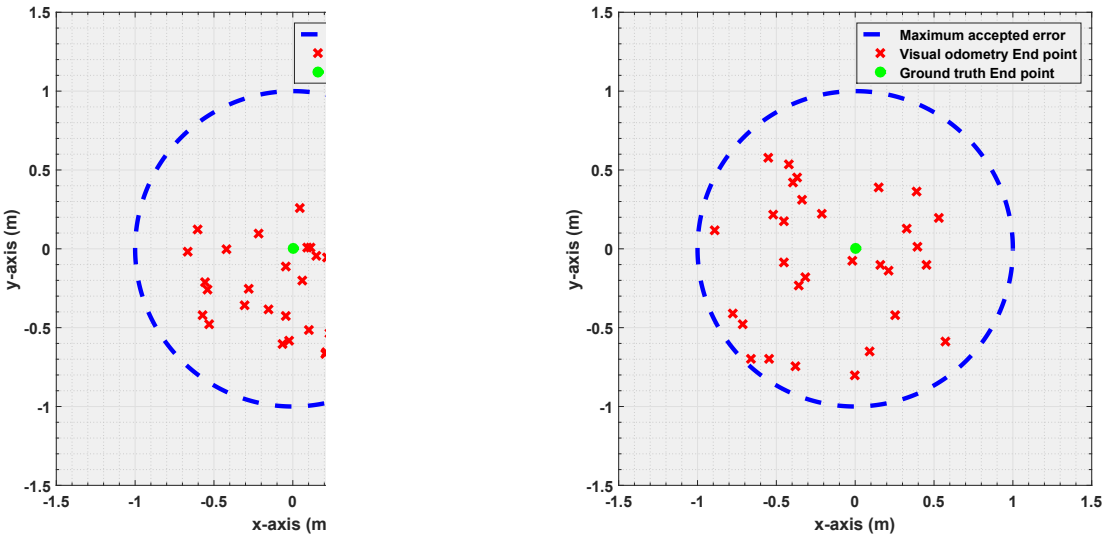
The obtained results form all experiments show the robustness of the proposed algorithm in estimating the pose of UAV. After comparing the results obtained from the experiments with different conditions, it has been found that in indoor flights, with higher density ground textures and better light conditions, the algorithm obtained more accurate results with lower accumulated error than the obtained in the outdoor flights.

Table 3.7 shows the comparison between the different flights environment with the total distance, the mean drift in each axis, the error in the final position, the



(a) (b) (c) (d)

Figure 3.26 Indoor flight: **a:** Translation vector (vision), **b:** Euler angles(vision), **c:** Pose estimation in 2D, and d) Pose estimation in 3D.



(a) Indoor (b) Outdoor

Figure 3.27 *End* point (final position) and the maximum error.

standard deviation and the total accuracy. The error in the final position was calculated by assuming the maximum error, that generates an error (100%) is located in a radius of 1 meter from the *End* point of the ground truth (Figure 3.27).

Table 3.7 Accuracy

Environment	**Indoor**	**Outdoor**
Number of Flights	30	30
Total Distance (m)	676.5	3742.1
Mean *(x,y)* Drift (m)	±0.21	±0.38
Mean *(z)* Drift (m)	±0.20	±0.30
Mean Position Error (%)	0.68	0.53
Max Position Error (%)	1.19	1.38
Standard Deviation	0.26	0.19
Accuracy (%)	99.32	99.47

Finally, Table 3.8 shows a comparison of the pose estimation accuracy between four approaches from the bibliography and the proposed algorithm. The results show that the best performance belongs to the proposed algorithm reaching a 99% of accuracy and a standard deviation of 0.23.

Table 3.8 Comparison of Visual Odometry Algorithms

Algorithm		**MD (m)**[a]	**MPE (%)**[b]	**SD** [c]	**Accuracy (%)**
Wang [308]	**VO**	14.88	—	8.87	—
	VO-IMU	1.98	—	1.16	—
Optical Flow	**Monocular [227]**	—	3.0	2.0	97.0
	Stereo [287]	±0.24	1.67	0.64	98.4
Proposed Algorithm		±0.29	0.61	0.23	99.4

[a] **MD** is the Mean *(x,y)* Drift (m). [b] **MPE** is the Mean Position Error (%).
[c] **SD** is the Standard Deviation.

Moreover, the estimated computation time is 64.2 *ms*. This result proved that the using SIFT-FREAK combination is quite faster than classical SIFT calculation, which obtained 110 *ms*.

3.6 Summary

This chapter presents a feature-based pose estimation algorithm using a UAV monocular downward-looking on-board camera. In this algorithm, a combination of the SIFT detector with the FREAK descriptor is used; in order to increase the performance and the robustness, and decrease the computational time of the feature points extraction and matching from the frame sequence, then the UAV pose (position and orientation) is estimated by calculating and decomposing the *frame-to-frame* and *world-to-frame* homographies of the matched features.

The obtained results proved the efficiency and the accuracy of the pose estimation algorithm that is based on visual data, with an overall performance of 50 cm $\times$ 30 cm of overall error for outdoor environment, and 35 cm $\times$ 20 cm in indoor scenarios on the *(x,y)-axes and z-axis* respectively. In the specific case of outdoor scenarios, where the conditions are more challenging, the presented system provided more accurate results than the DGPS which suffered from problems due to the signal degradation.

The results obtained for both indoor and outdoor scenarios prove the viability of the visual based navigation system for complementing the classical localization algorithms, even in the most challenging long term missions, where the system proved its viability.

In addition, the results of the computational time proved that combining the SIFT detector and FREAK descriptor yields a robust and fast detection algorithm, witch ia able to provide real-time navigation system based on visual information.

CHAPTER 4

Monocular Vision-based Obstacle Detection

One of the most challenging problems in the domain of autonomous aerial vehicles is the designing of a robust real-time obstacle detection and avoidance system [1]. This problem is getting more complex especially for SUAVs and MAVs, because of the limited number of the on-board sensors due to the SWaP constraints. Therefore, using lightweight sensors; such as Complementary Metal-Oxide-Semiconductor (CMOS) cameras, can be the best choice comparing with

[1] Publications of the author related to the chapter:

- Size-Expansion Monocular Vision-Based Obstacle Detection/Avoidance for Unmanned Aerial Vehicles [15].
- Obstacle Detection and Avoidance System based on Monocular Camera and Size Expansion algorithm for UAVs[13].

other sensors; such as laser [197, 276, 279] or radar [27].

For real-time applications, different works focus on using stereo cameras to obtain and reconstruct the 3D model of the obstacles, or to estimate their depth. Instead, this chapter proposed a bio-inspired approach that mimics the human behavior of detecting the collision state of the approaching obstacles by realizing its size expansion [114] using monocular camera. During the UAV motion, the detection algorithm estimates the changes in the size of the area of the approaching obstacles.

The system is divided into two main stages: The **guidance stage**, at which the obstacle detection algorithm is performed based on the input images that are captured by the frontal camera. The second stage is the **motion control**, where the avoidance decision is taken and is sent to the UAV. Figure 4.1 shows the general

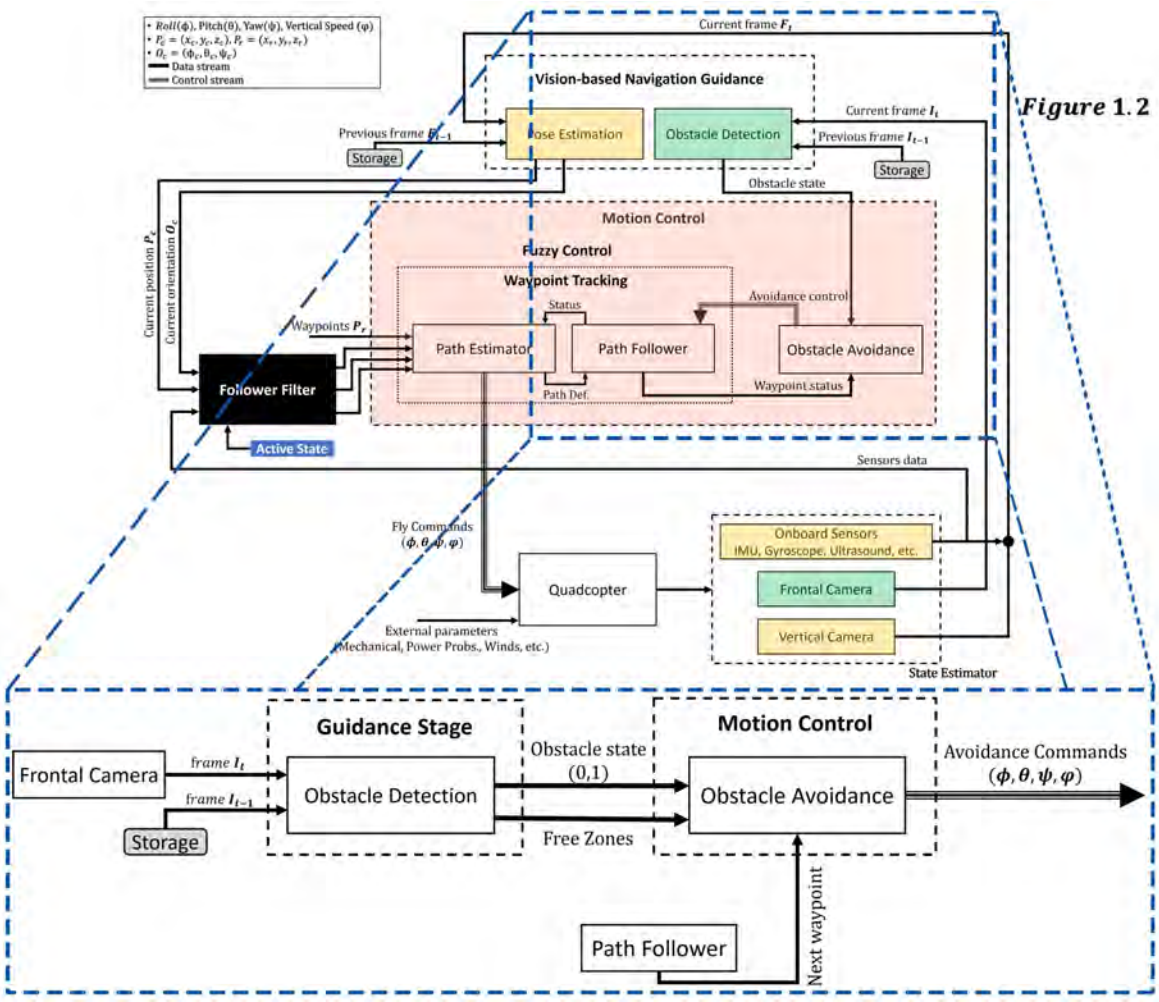


Figure 4.1 General overview of the Detection and Avoidance phases

overview of the proposed system. First, the method detects the feature points of the obstacles, then extracts the obstacles that have the highest probability of getting close toward the UAV. Secondly, by comparing the area ratio of the obstacle in the frame sequence, and the position of the UAV, the method decides if the detected obstacle may cause a collision or not. Finally, by estimating the obstacle 2D position in the image, and combining with the tracked waypoints, the UAV performs the avoidance maneuver and returns back to its path.

4.1 Obstacle Detection

The proposed obstacle detection algorithm mimics the human behavior of detecting the obstacles that are located in front of the UAV during motion. At which point, the collision state of the approaching obstacles is estimated instead of building 3D models, or calculating the depth of the obstacle.

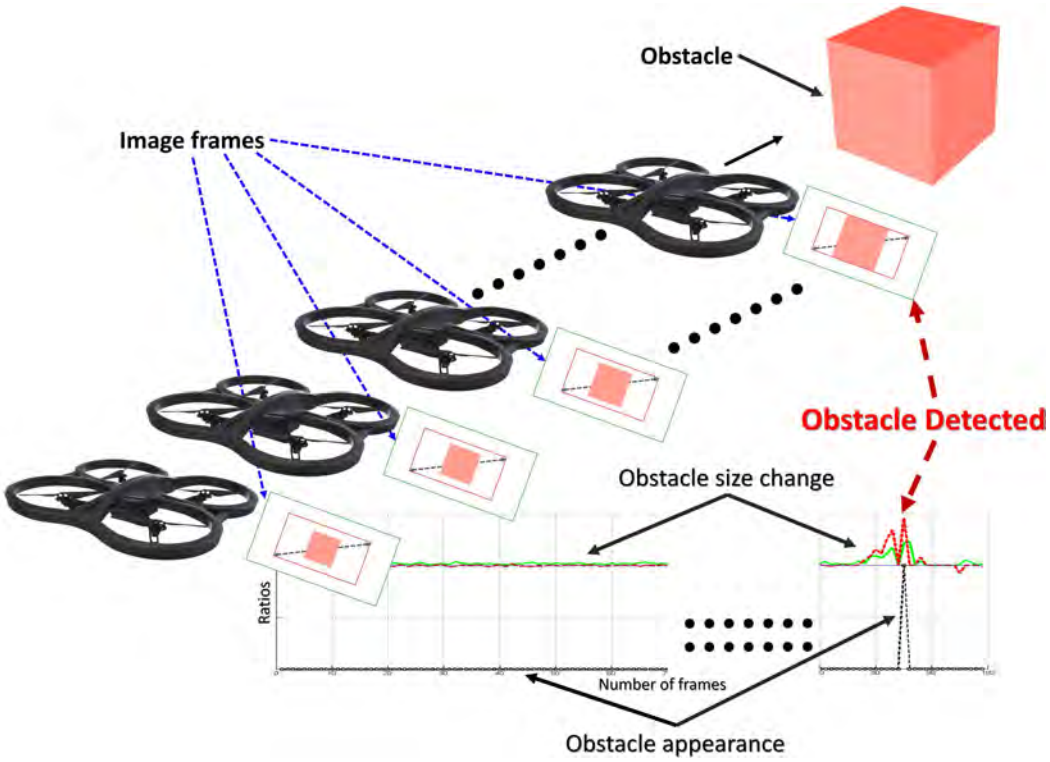


Figure 4.2 The concept of approaching obstacle detection

The novelty and the key of this algorithm is to estimate the size ratios of the approaching obstacles from the consecutive frames during the flight as shown in

Figure 4.2. This is achieved by estimating the change in the size property of the detected feature points (diameter), and the size of the convex hull (area) which is constructed from these points as well. When the size ratios exceed certain empirical values(explained in 4.1.2), it means that there is an obstacle detected, and can cause a danger to the UAV as shown in Algorithm 3, and Figure 4.3.

Algorithm 3: Obstacle Detection

Input: Input frames F
Output: Collision state Obs, Collision-free zone τ
Define: Current frame F_t, Previous frame F_{t-1}, Current workspace ROI_t, Previous workspace ROI_{t-1}, Current keypoints $kp_t[\,]$, Previous keypoints $kp_{t-1}[\,]$, Number of keypoints (N, M), $n \in N$, $m \in M$, $n = m$, $x \in n$, Matched points pts, Distance ratio $thresh$, Object of interest $Convex$
begin
  while *isFlying( )* **do**
    $F_{t-1} \leftarrow$ getNewFrame( )
    if *Obstacle Detection isActivated( )* **then**
      $F_t \leftarrow$ getNewFrame( )
      $(ROI_{t-1}, ROI_t) \leftarrow$ DefWrkspc(F_{t-1},F_t) // Extract 62^o ROI
      $(kp_{t-1}(N), kp_t(M)) \leftarrow$ DetectKeypoints(F_{t-1}, F_t)
      $(pts_{t-1}(n), pts_t(m)) \leftarrow$ MatchSymKeypoints($kp_{t-1}(N)$, $kp_t(M)$, $thresh$) // See Algorithm 4
      $(pts_{t-1}(x), pts_t(x)) \iff \text{size}(pts_t(m)) > \text{size}(pts_{t-1}(n))$
      // Return points with bigger size
      $(Convex_1, Convex_2) \leftarrow$ CreateOOI($pts_{t-1}(x)$, $pts_t(x)$)
      if $KPSizeRatio(pts_t(x) : pts_{t-1}(x) \geq 1.2)$ *and* $ObAreaRatio(Convex_2 : Convex_1) \geq 1.7$ **then** // Compare size ratios
        $Obs \leftarrow$ true
        $\tau \leftarrow$ CalcPos(Obs) // Estimate collision-free zones
        goto Algorithm 7 // See Algorithm 7 (Chapter 5)
      end
    end
    $F_{t-1} \leftarrow F_t$
  end
end

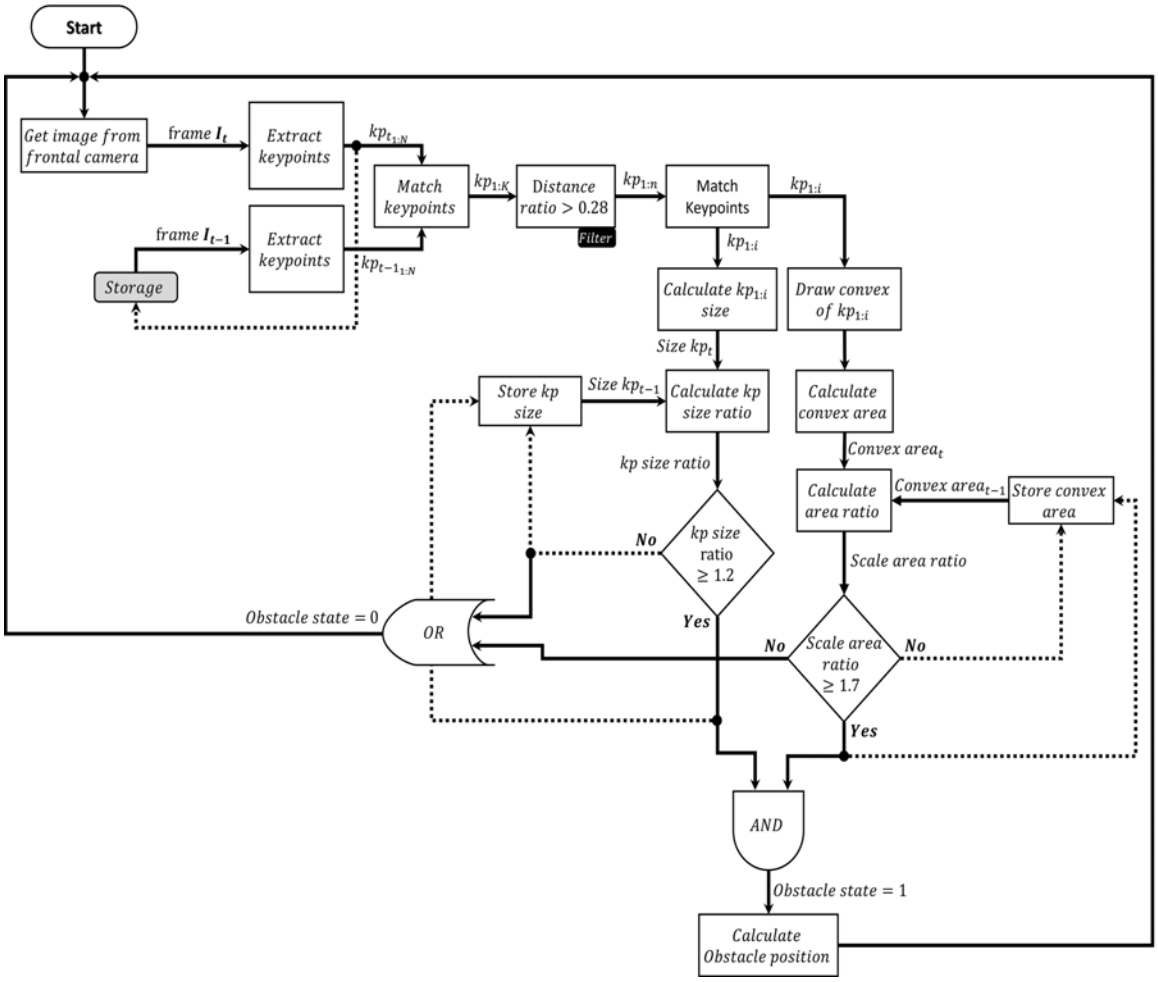


Figure 4.3 Obstacle detection approach flowchart

4.1.1 Feature Detection and Description

In this step, an image Region Of Interest (ROI) of 62^o diagonal FOV is taken, in order to be processed instead of the whole image, as it shown in Figure 4.4. The selection of the diagonal 62^o ROI is based on the results that are obtained from the experiments. Where, it has been found that any object is detected out of the area of this ROI will not cause any danger to the UAV, and only the objects that are detected in the scope of this diagonal 62^o ROI can be considered as an obstacle. Furthermore, processing the diagonal 62^o ROI instead of the whole diagonal 92^o image, leads to a significant minimizing in computational time. The carried out tests proved the viability of this approach, and the results will be discussed in following sections.

Due to flying in unknown environments structures, the captured frames are af-

fected with different conditions; such as illumination variation which may induce to noise and error. However, the keypoints need to be extracted accurately even under these conditions. Therefore the SIFT detector algorithm is used; because of the ability to identify and localize accurately the feature points even under different image conditions especially scale and rotation properties.

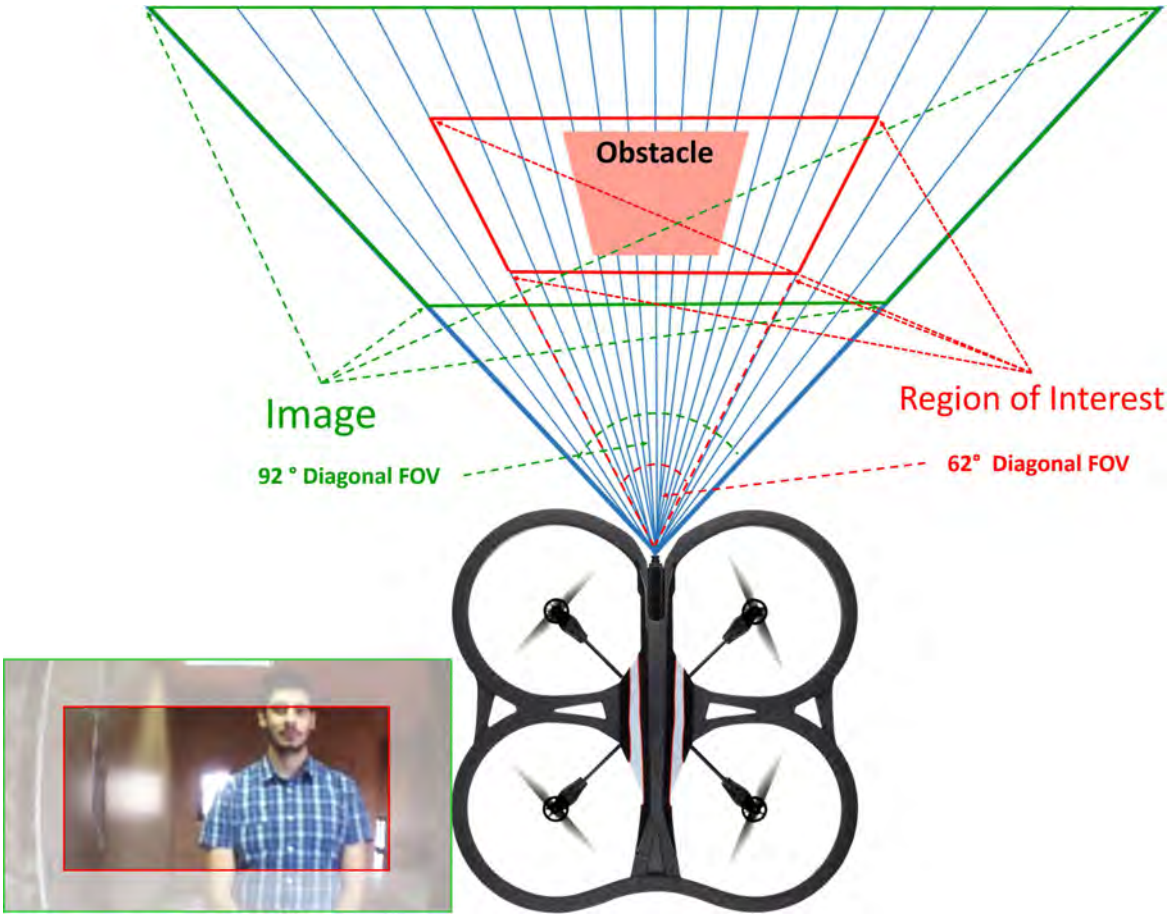


Figure 4.4 Define the diagonal 62^o patch from the whole 92^o image FOV

According to Algorithm 3, all the keypoints are detected and its descriptors are extracted from the two consecutive frames as shown in Figure 4.5, then a vector of the position (x, y) and the size (s) of each keypoint is obtained.

After detection the keypoints, the $Brute - Force$ algorithm is applied to match the keypoints from the two frames, and only the points that are found in both frames are returned. Algorithm 4 illustrates the concept of the $Brute - Force$ method to match, and find the smallest distance of a pair of points.

For more accuracy, the matched keypoints are filtered, by eliminating the ones

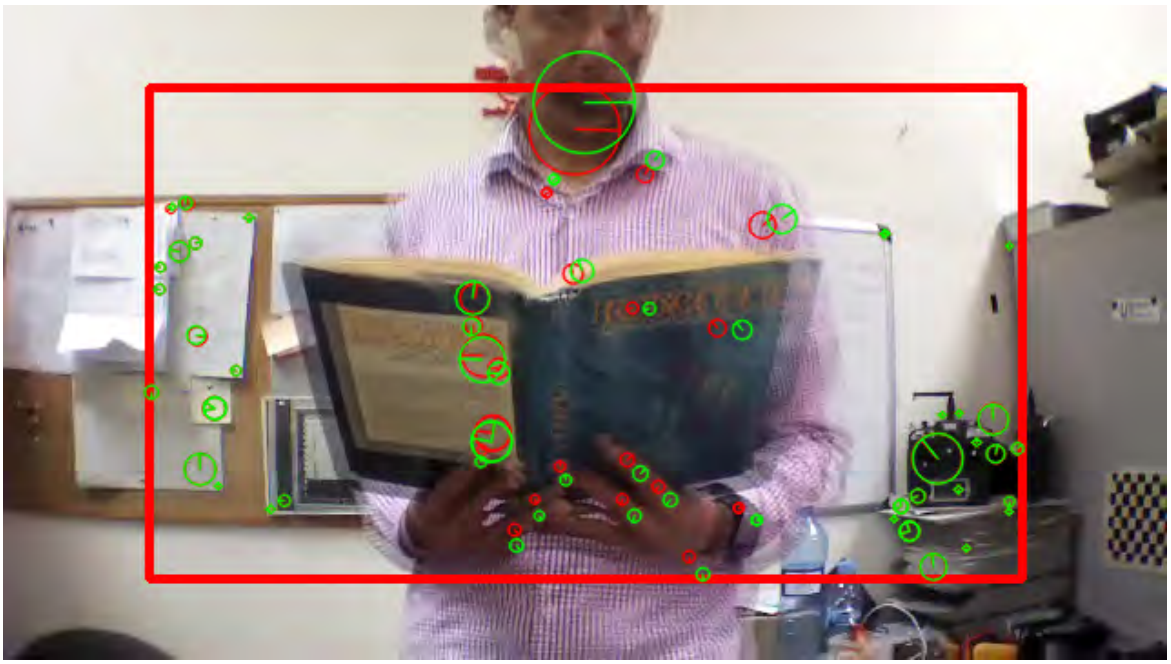


Figure 4.5 Keypoints extraction from two consecutive frames; keypoints extracted from frame f_{t-1} (***red***) and keypoints extracted from frame f_t (***green***)

Algorithm 4: Brute-Force Matcher

Input: Array of keypoints in first image $kp_{t-1}[N]$,
Array of keypoints in second image $kp_t[M]$
Output: Indices of the matched points $index_1$, $index_2$

Define: Keypoints $kp_1 = (x_1, y_1), \ldots\ldots, kp_n = (x_n, y_n)$, Minimum distance d_{min}, Distance between keypoints d

begin
  $d_min \leftarrow \infty$
  for $i \leftarrow 1, N$ **do**
    for $j \leftarrow i + 1, N$ **do**
      $d \leftarrow \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$
      if $d < d_{min}$ **then**
        $d_{min} \leftarrow d$
        $index_1 \leftarrow i$
        $index_2 \leftarrow j$
      end
    end
  end
end

that have a minimum distance ratio bigger than an empirical threshold value (0.28). Let mkp is the filtered-matched keypoint which are calculated as follows:

$$mkp(n) = \begin{cases} (x, y, s), & distanceratio \leq 0.28 \\ 0, & otherwise \end{cases} \qquad \forall n \in K \tag{4.1}$$

where, s is the size of the keypoint (diameter), $distanceratio$ is the minimum distance ratio of the matched keypoints, and K is the total number of matched keypoints.

Afterwards, the obtained keypoints by Equation 4.1 are compared from the second to the first frame, and then the algorithm return the matched keypoints **if and only if** its size is growing, as shown in Figure 4.6.

$$mkp(i) = \begin{cases} (x, y, s), & Size(mkp_2(i) > mkp_1(i)) \\ 0, & otherwise \end{cases} \qquad \forall i \in n \tag{4.2}$$

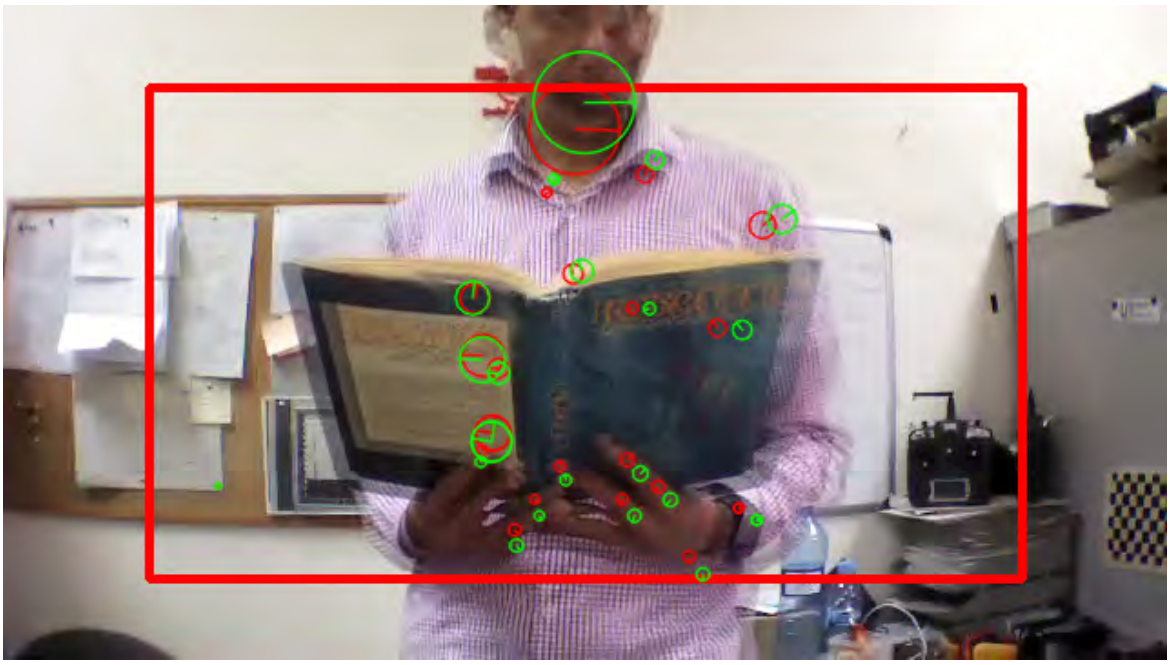


Figure 4.6 Filtered keypoints where the size expand from the second frame to the first frame; keypoints extracted from frame f_{t-1} (***red***) and keypoints extracted from frame f_t (***green***)

4.1.2 Object of Interest (OOI)

The next step of the detection algorithm is to determine the probability to detect a frontal obstacle. Hence, from the extracted and filtered keypoints by Equation 4.2, an Object of Interest (OOI) is created around these keypoints in both frames, by creating a convex hull of the corresponding points, as it is shown in Figure 4.7, and is calculated as follows:

$$C = \sum_{i=1}^{N} \lambda_i mpk_i | (\forall i : \lambda_i \geq 0) \tag{4.3}$$

where C defines the convex hull, and λ_i is a non-negative weight assigned to the keypoints $mpk_i \in N$ and $\sum_{i=1}^{N} \lambda_i = 1$.

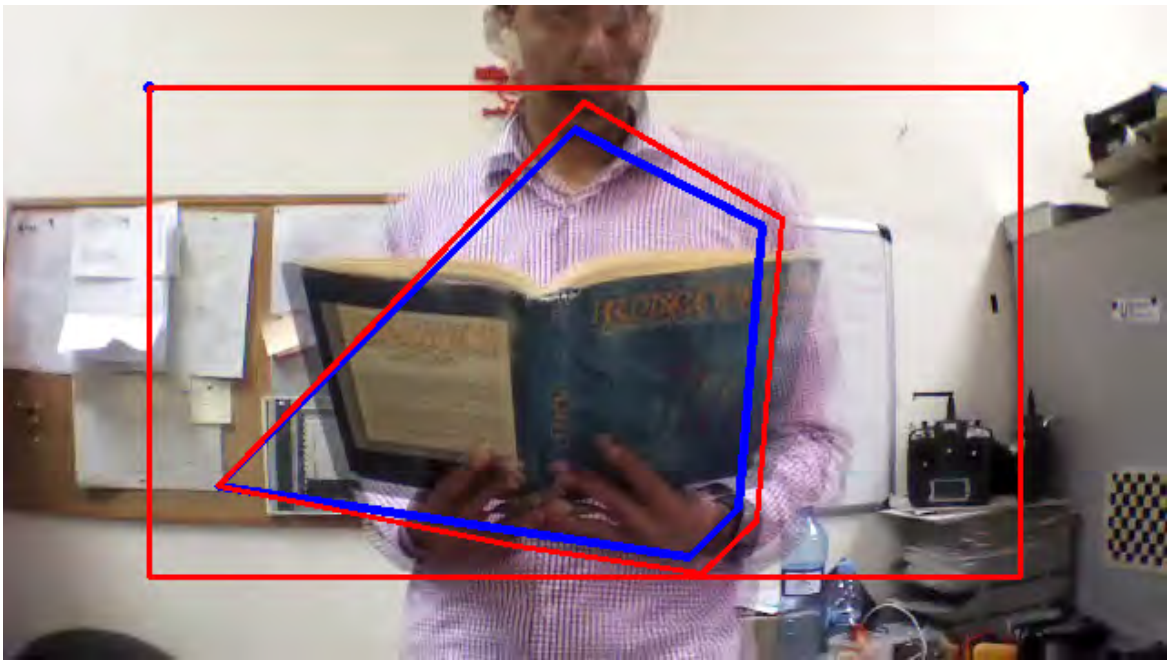


Figure 4.7 Convex Hull construction from detected keypoints in both frames; frame f_{t-1} (**blue**) and f_t (**red**)

Next, in order to estimate the changes in the size of the area of the detected obstacles, it is considered that each convex hull as an irregular polygon. Therefore,

for a given C as a convex hull, the area of C can be calculated as follows:

$$C_{area} = \frac{1}{2}\begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ \vdots & \vdots \\ x_n & y_n \\ x_1 & y_1 \end{vmatrix} = \frac{1}{2} \begin{matrix} [(x_1y_2 + x_2y_3 + x_3y_4 + \cdots + x_ny_1) \\ -(y_1x_2 + y_2x_3 + y_3x_4 + \cdots + y_nx_1)] \end{matrix} \tag{4.4}$$

where $x_{(1:n)}$ and $y_{(1:n)}$ are vertices, and n is the number of sides of the polygon.

Finally, the size ratio of the matched keypoints, and the area of the convex hull from the second to the first frame are calculated respectively as follows:

$$ratio(mkp) = \frac{1}{N}\sum_{i=1}^{N}\frac{Size(mkp_2(i))}{Size(mkp_1(i))} \tag{4.5}$$

$$ratio(C) = \frac{Size(C_2)}{Size(C_1)} \tag{4.6}$$

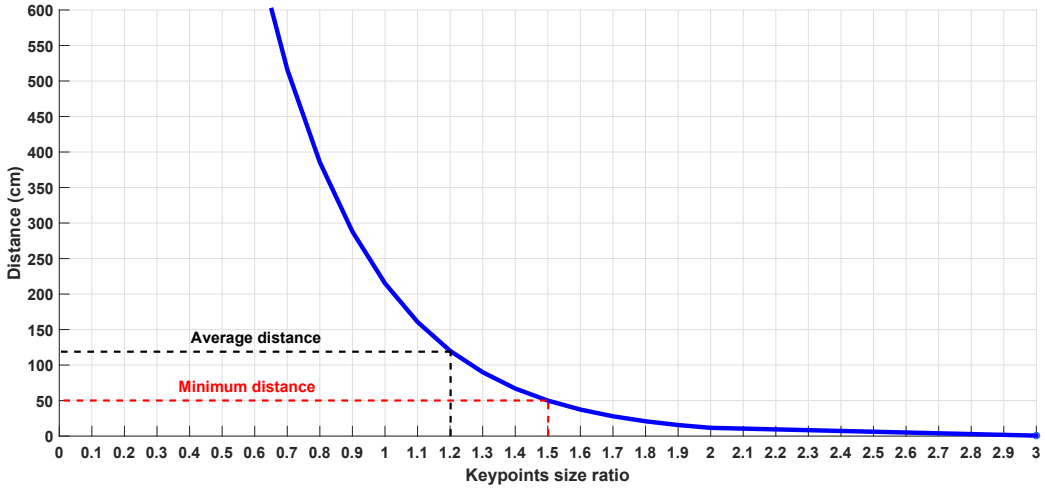
Thereafter, the algorithm estimates the collision state, if the approaching obstacle may represent an collision or not.

$$State = \begin{cases} 1, & ratio(mkp) \geq 1.2 \bigwedge ratio(C) \geq 1.7 \\ 0, & otherwise \end{cases} \tag{4.7}$$

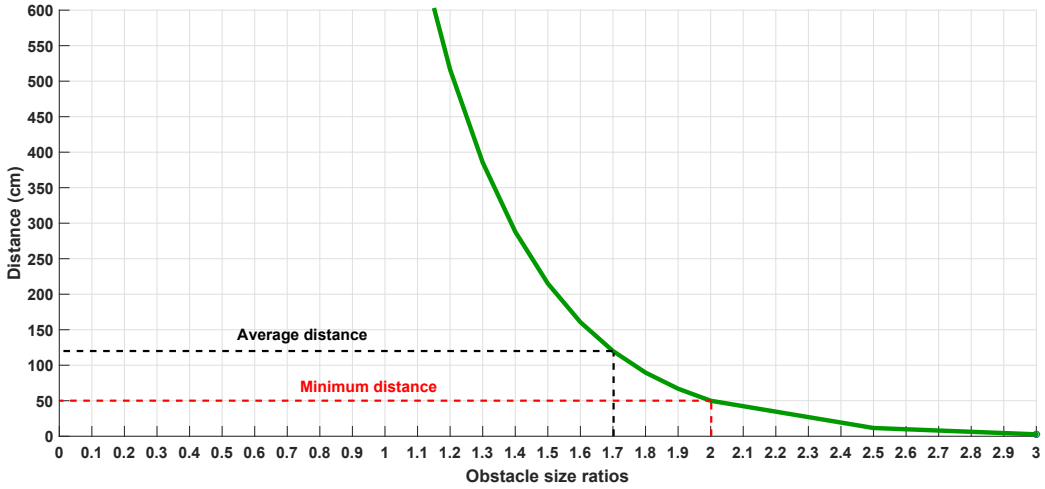
Next, an empirical study about the relation of the ratios between the size of the keypoints, the area of the obstacle and the distance of the approaching obstacle has been developed and the results are illustrated in Figure 4.8. This relation has been estimated by performing different indoor and outdoor experiments. Assuming that the UAV is flying at a constant velocity, the best ratios are in the range of [1.2–1.5], and [1.7–2.0] for keypoints size and obstacle size area respectively, at which the obstacle can be detected in a distance of [120–50] cm.

Figure 4.9 shows the collision state of the detected obstacles by the monocular camera, where it provides **1** if there is an obstacle, or it provides **0** if there is no obstacle detected.

In this step, after detecting the obstacles with a collision state value 1, the algorithm estimates the position of the extremely outer points that construct the obstacle in the image (P_l, P_r, P_u, P_d), as it is shown in Figure 4.10, where P_l is the point



(a) Keypoints.



(b) Obstacle Area.

Figure 4.8 The relation between the distance and the size ratios

the of a position that has the minimum x value, P_r has the maximum x value, and similarly, P_u and P_d have the y minimum and maximum values respectively.

Finally, the collision-free zones $Left$, $Right$, Up and $Down$ (in case of hanged or flying obstacles) are calculated as four rectangles surrounding the obstacle as shown en Equation 4.8:

$$\begin{aligned} \tau &= \left(\tau_l, \quad \tau_r, \quad \tau_u, \quad \tau_d\right) \\ &= \left(Zone_{L_{width}}, \quad Zone_{R_{width}}, \quad Zone_{U_{height}}, \quad Zone_{D_{height}}\right) \end{aligned} \tag{4.8}$$

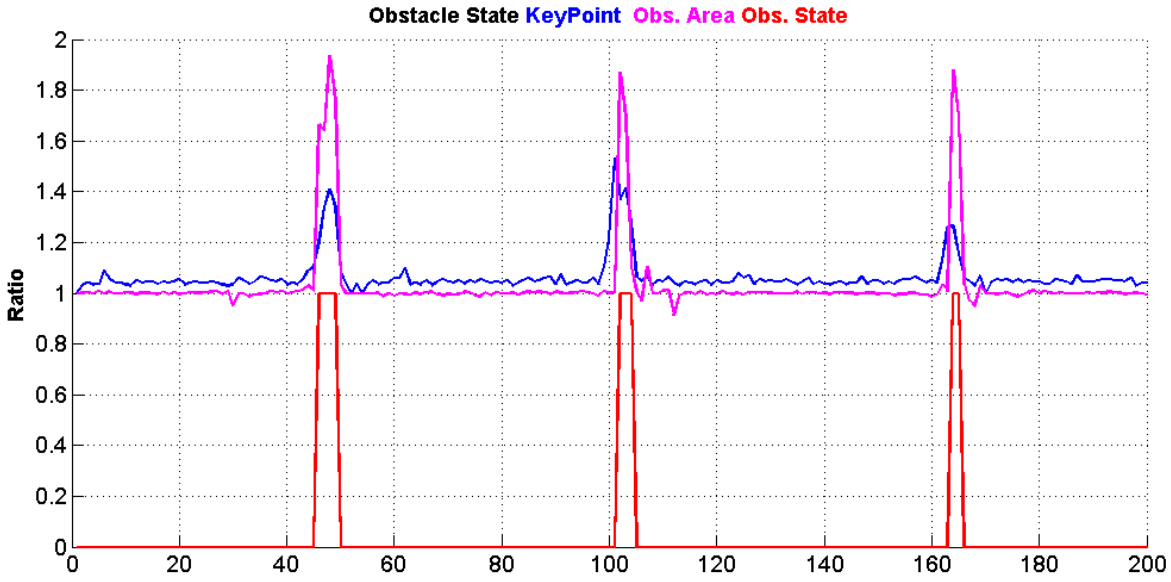


Figure 4.9 Obstacle State: **Blue:** Keypoint size ratio, **Magenta:** Convex area ratio and **Red:** Obstacle State **(0)** not fount **(1)** found.

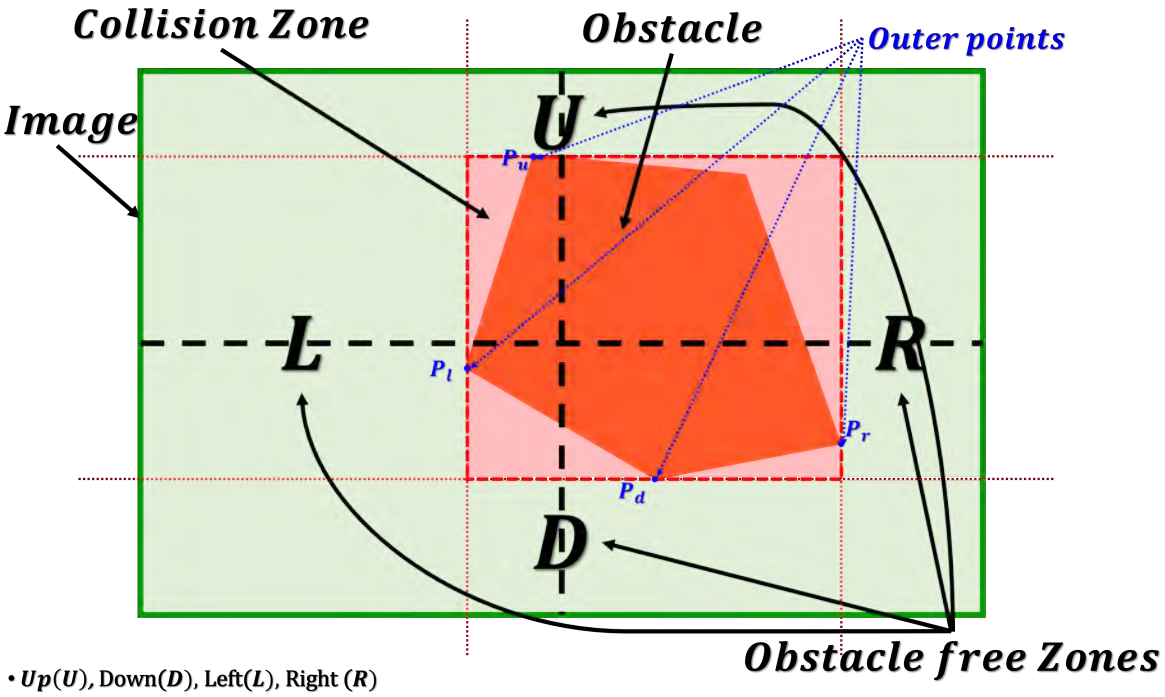


Figure 4.10 Estimating Obstacle outer points.

where, $Zone_{L_{width}}$, $Zone_{R_{width}}$, $Zone_{U_{height}}$ and $Zone_{D_{height}}$ are the width and the height

of the rectangles that are created by the points (P_l, P_r, P_u, P_d), as follows:

$$\begin{aligned} Zone_L &= Rectangle\left[(0, P_{u_y}),\ (P_{l_x}, P_{u_y}),\ (0, P_{d_y}),\ (P_{l_x}, P_{d_y})\right] \\ Zone_R &= Rectangle\left[(P_{r_x}, P_{u_y}),\ (ROI_w, P_{u_y}),\ (P_{r_x}, P_{d_y}),\ (ROI_w, P_{d_y})\right] \\ Zone_U &= Rectangle\left[(P_{l_x}, 0),\ (P_{r_x}, 0),\ (P_{l_x}, P_{u_y}),\ (P_{r_x}, P_{u_y})\right] \\ Zone_D &= Rectangle\left[(P_{l_x}, P_{d_y}),\ (P_{r_x}, P_{d_y}),\ (P_{l_x}, ROI_h),\ (P_{r_x}, ROI_h)\right] \end{aligned} \tag{4.9}$$

where, w and h are the width and the height of the ROI respectively.

4.2 Experimental Results

In order to evaluate the performance of the proposed detection algorithm, different experiments have been carried out of 100 real flights using the Parrot Ar.Drone 2.0, in both indoor and outdoor environments, with a total number of 1000 obstacles, taking in consideration the visual conditions (the illumination and the texture of the obstacles) which affect the accuracy of the detection.

Similar to the experiments in Chapter 3 of validating the pose estimation algorithm, the ground station has an Intel i7-3770 at 3.4 GHz CPU, with 6 GB DDR3 RAM, and the connection with the UAV is established via a standard 802.11n wireless LAN card.

4.2.1 Scenarios

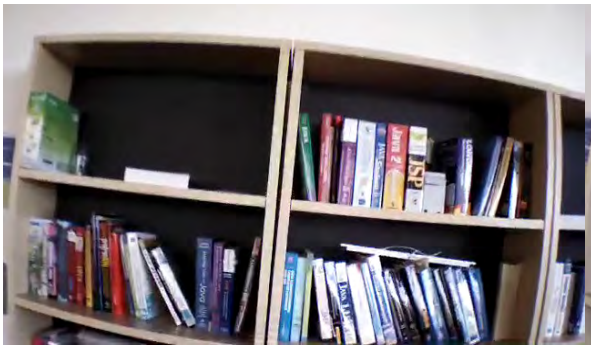
Two different scenarios have been conducted, in order to evaluate the performance of the proposed algorithm, with data gathered from the experiments to test the detection and the estimation of the position of the obstacle. In each scenario, different types of obstacles (people, obstacles, pillars, trees and walls), (statics and dynamic) are situated.

The first scenario is a predefined straight flight, where the UAV flies in a straight line from the starting point to the end point. Different types of obstacles with unknown previous position were situated in the UAV path. The goal of this scenario is to evaluate the accuracy, and robustness in detecting, and avoiding the obstacles in motion.

The second scenario is a hover stability flight. At which, the UAV enters to the hover flight mode, and different obstacle are approaching to it. Once the obstacle is detected, the UAV flies in the the opposite direction of the obstacle (*Backward maneuver*).

4.2.2 Results

From the experiments, the obtained results demonstrate that the algorithm is able to detect the obstacles with different sizes (areas) between 8500 and 200000 pixels, and at a distance range between 50 and 120 *cm*. It is shown that the minimum accuracy of the algorithm is 95.0%, and the overall accuracy is 97.4% as it is demonstrated in Table 4.1.



(a) Input frame F_t.

(b) Input frame F_{t+1}.

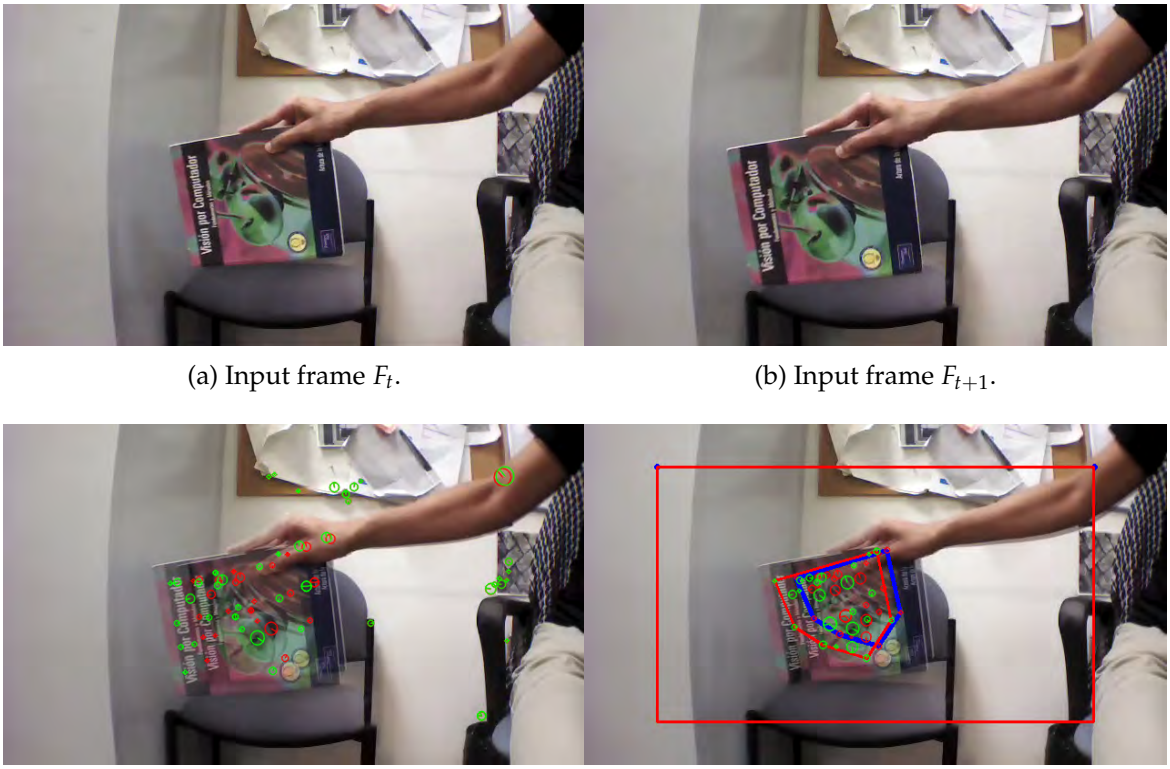
(c) Keypoints within the 62^o FOV.

(d) Detected obstacle.

Figure 4.11 Obstacle detection: $ratio(mkp) = 1.27$, $ratio(C) = 1.76$ and $distance = 114$ *cm*

Figures 4.11, 4.12 and 4.13 illustrate the detection process of various approaching obstacles, with different size ratios. Where, 4.11a, 4.12a, 4.13a and 4.11b, 4.12b, 4.13b are showing the two input consecutive frames to be processed. In 4.11c, 4.12c

or 4.13c it is shown the total number of the detected and matched keypoints before filtering its size expansion property. Finally, the filtered keypoints and the constructed polygon of the detected obstacle are shown in 4.11d, 4.12d and 4.13d.



(a) Input frame F_t.

(b) Input frame F_{t+1}.

(c) Keypoints within the 62^o FOV.

(d) Detected obstacle.

Figure 4.12 Obstacle detection: $ratio(mkp) = 1.25$, $ratio(C) = 1.71$ and $distance = 92$ cm

Table 4.1 summarizes the accuracy of the detection algorithm. The table shows the total number of the obstacles that either situated in the UAV path (*first scenario*) or moving towards the UAV (*second scenario*), the number of the detected obstacles and the number of fails.

From the table, it is illustrated that the accuracy of the detection process in the indoor scenarios is better than the accuracy in outdoor environments. This is due to the constancy of the light conditions in indoors rather than outdoors, which are suffered from various lighting effects.

Two main reasons for the fail of detection; the first one is the disability of extracting sufficient number of keypoints, and that is either because of the low light conditions or because of the absence of the texture on the obstacle surfaces; such as

(a) Input frame F_t.

(b) Input frame F_{t+1}.

(c) Keypoints within the 62^o FOV.

(d) Detected obstacle.

Figure 4.13 Obstacle detection: $ratio(mkp) = 1.20$, $ratio(C) = 2.15$ and $distance = 126$ cm

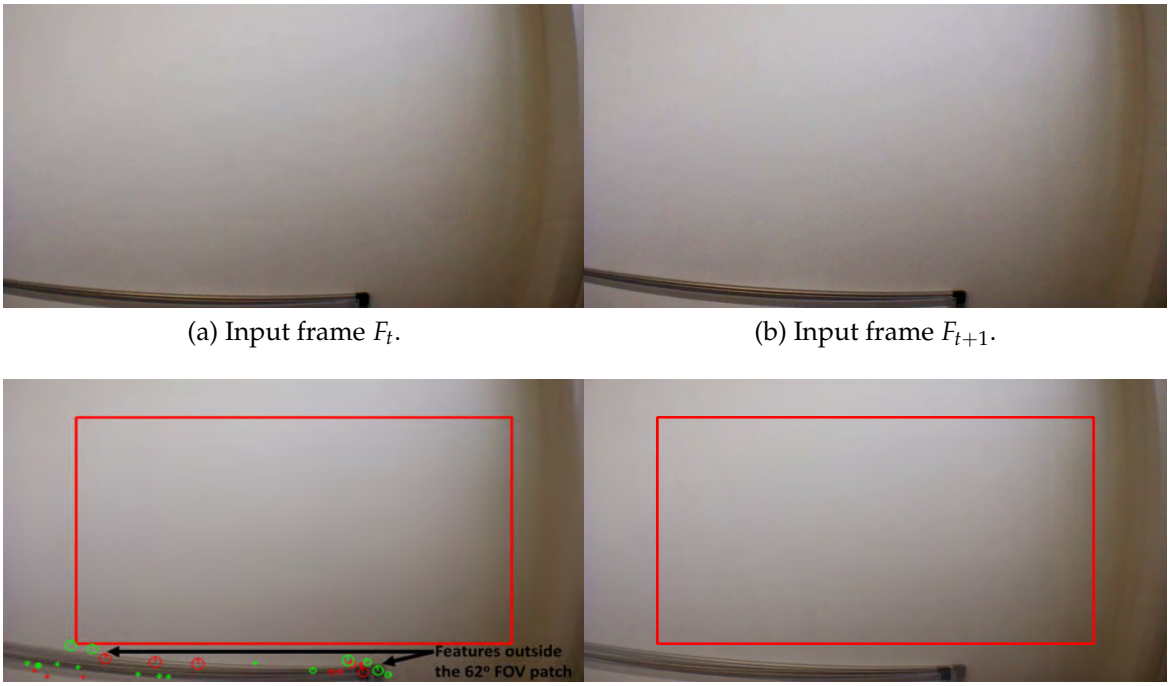
Table 4.1 Accuracy of Detection Algorithm

	Indoor				**Outdoor**				
	People	**Obstacle**	**Pillar**	**Wall**	**People**	**Obstacle**	**Tree**	**Wall**	**Total**
Situated	200	110	80	80	200	120	140	70	1000
Detected	196	107	79	76	196	116	135	69	974
Fail	4	3	1	4	4	4	5	1	26
Accuracy (%)									
Object	**98.0**	**97.3**	**98.8**	**95.0**	**98.0**	**96.7**	**96.4**	**98.6**	**97.4**
Environment	**97.3**				**97.4**				

in the case of some pillars and walls as shown in Figure 4.14.

The second reason is the direction of the motion of the obstacle, the algorithm is able to detect the moving obstacle if the motion is towards the UAV (approaching obstacles).

Figure 4.15 shows an example of the second scenario, where the UAV flies in



(a) Input frame F_t.

(b) Input frame F_{t+1}.

(c) Keypoints within the 62^o FOV.

(d) Detected obstacle.

Figure 4.14 Obstacle detection fail (wall) (absence of texture): $ratio(mkp) = 1$ and $ratio(C) = 1$

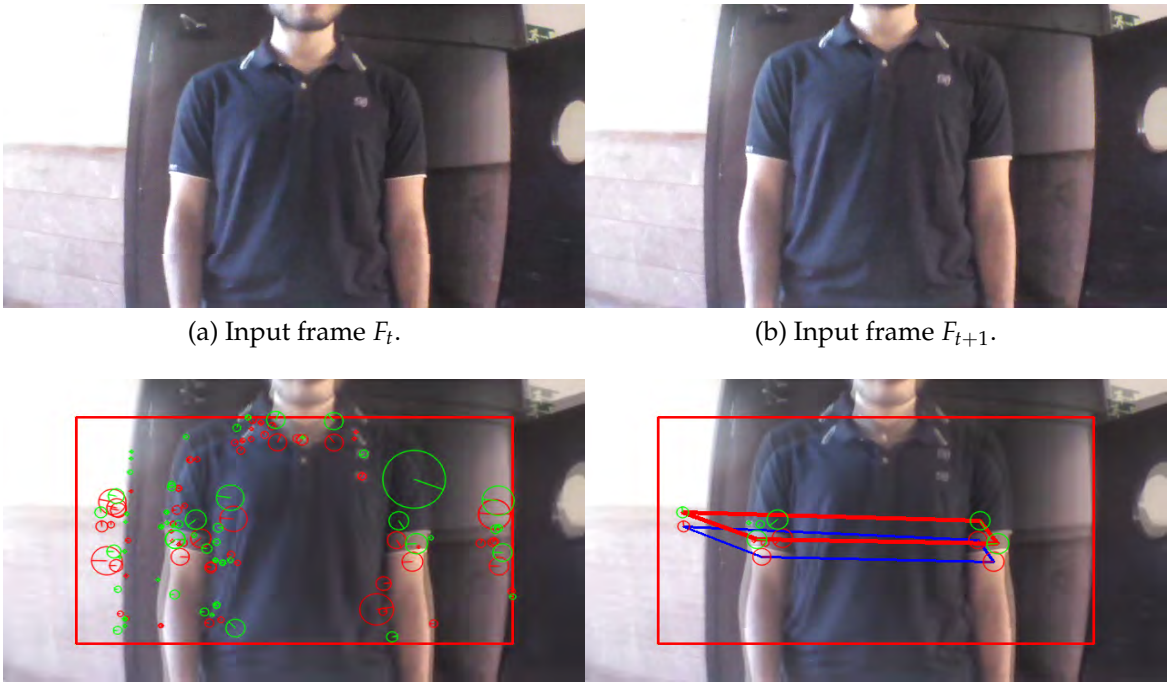
hover mode, and the object is moving, however, this movement is not in the direction of the UAV. Therefore, it does not consider as an obstacle.

However, in most cases of the moving obstacles according to Table 4.1, the algorithm could not detect the appearance of the obstacles if the motion is around the UAV and not approaching to it; such as in the case of the people and obstacles.

In addition, the proposed algorithm is evaluated against two related works of detecting frontal obstacles based on monocular vision. As it shown in Table 4.2, the proposed algorithm provides more accuracy (97,4%) comparing to *SURF + Template matching* method [228] which provides 97%, and *relative distance estimation* approach [261] that provides 97.1% of accuracy.

Table 4.2 Comparison of Frontal Obstacle Detection

Algorithm	**Total**	**Detected**	**Fail**	**Accuracy (%)**
SURF + Template matching [228]	107	104	3	97
relative distance estimation [261]	35	34	1	97.1
Proposed Algorithm	1000	974	26	97.4



(a) Input frame F_t.

(b) Input frame F_{t+1}.

(c) Keypoints within the 62^o FOV.

(d) Detected obstacle.

Figure 4.15 Obstacle detection fail (people) (motion around the UAV): $ratio(mkp) = 1.07$ and $ratio(C) = 1.03$

Furthermore, The computational time of the detection algorithm is estimated around $52.4ms$. This is due to the processing ROIs of 62^o FOV, which leads to decrease the processing time up to 50% from $106.1ms$ comparing to processing the whole 92^o FOV images. In addition, this computational time is estimated for the detection of $800 - 1200$ keypoints. However, if the number of detected keypoints exceeds 6000, the computational time peaks to a maximum of $100ms$, on the other hand, if it is below 300 keypoints, then the required computation time is reduced to $30ms$.

4.3 Summary

In this chapter, an algorithm has been presented as a framework to cope with cutting-edge UAVs technology. Real-time obstacle detection is studied as complex and essential task for intelligent aerial vehicles in transportation systems. The proposed algorithm take the advantages of on-board camera to accomplish complex tasks, that is, safe obstacle sensing and detection tasks.

The selected configuration of a diagonal 62° FOV ensured the capabilities of detecting the border of an object of the size of the actual UAV (58.4 $\times$ 1.3 $\times$ 54.4) located in the center of the image at distances higher the 15 cm, which allows to avoid obstacles, for higher obstacles or closer distances. Bigger obstacles located at longer distances were avoided due to the use a high quality camera able to detect obstacle at long distances. In case of faster speeds required, the frame rate calculation and the angle should be adjusted, to allow the UAV to do the calculation at proper detection. However, the change of the field of view of the camera, would only be advisable in order to allow further maneuverability in extremely dense scenarios, with short distance detection requirements which are not common in aerial scenarios where UAVs are deployed.

Keypoints-used approach is based on the use of SIFT features. The performance obtained proved to be good in both computational time and overall detection performance scheme. The nature of the approach made it possible to adapt it to different set of features beyond the use of SIFT; such as the FAST-BRIEF pair [273] and BRISK [143] which proved to provide better performance in different scenarios. Future works will try to analyze the advantages of adding these sets of features to the presented approach.

The usefulness and advantages of the presented reliable solutions are demonstrated through real results under demanding circumstances, such as, complex obstacles and close pedestrians. Hence, complex scenarios are evaluated and difficulties are successfully overcame by means of monocular camera processing, where the relative size expansion of the obstacles are estimated and the approaching obstacles are detected from a distance between 90 and 120 *cm* with 97.4% of total accuracy. The various performed tests proved both, the trustable performance of the algorithms provided and the improvements in comparison to the previous works presented in literature.

CHAPTER 5

Visual Servoing for UAVs

In the previous chapters, vision-based algorithms have been presented, in order to achieve several tasks that are required for the autonomous UAVs. These algorithms covered the areas of the estimating the position and orientation of the UAV, as well as the sensing and detecting the obstacles that are situated in the UAV path.

This chapter [1] aims to include the data obtained from the proposed vision-based algorithms as an input to the control loop of the UAV, in order to perform the flight maneuvers for the autonomous navigation purposes. Based on the processed information and the type of the task, two control systems have been presented (FLC

[1] Publications of the author related to the chapter:

- Autonomous Indoor Navigation of Low-Cost Quadcopters [140].
- Waypoint Tracking, Guidance and Control for Autonomous Unmanned Aerial Vehicles (UAVs), Journal of Control Engineering Practice, 2017. (Under revision).

and PID), that can be used for the waypoint tracking and the collision avoidance purposes.

5.1 Monocular vision-based waypoint tracking

One of the main objectives of the autonomous UAVs is the ability to fly safely from the start point to the destination. To achieve this purpose, a robust guidance algorithm in order to generate the flight path based on waypoint definitions (*Path Planning*), a strategy of following and swapping the generated waypoints efficiently (*Waypoint Tracking*), and a stable control system to perform the path following as well as the obstacle avoidance maneuver are required.

5.1.1 Path planning

In this dissertation, the path planning algorithm is based on the approach presented in [141]. At which, based on the grid maps of static environments, the algorithm finds the solution of the best path in terms of the minimum distances. In addition, to optimize the best path, the Simulated Annealing (SA) optimization algorithm has been used because it provided the best results [185, 352].

SA is a probabilistic random search technique, which is used to solve large-scale problems for approximating the global optimum, that is hidden among many local optima. It is a metaheuristic algorithm of the trajectory-based family, which is the set of approaches, that uses a single solution throughout the algorithm in order to find the optimal solution.

The concept of SA is inspired by the physical annealing process [284]; where the temperature is used as a control parameter with the cooling schedule, to search for the global optimum, using the concept of probabilistically accepting non-improving solutions, to avoid being trapped in local optima.

The algorithm takes the *grid map*, *start* point and the *destination* point as inputs. At each iteration, a solution must be generated to be evaluated, starting with a random solution. After generating the first solution, the feasibility of this solution, must be checked in terms of reaching the destination point, and not stuck in deadened path.

As it is shown in Figure 5.1, the grid map is built of three types of cells, where **white** cells presents the free space, **black** cells represent the obstacles, **gray** cells

are the safety boundary areas, **green** cell is the start point and the **red** cell is the destination point.

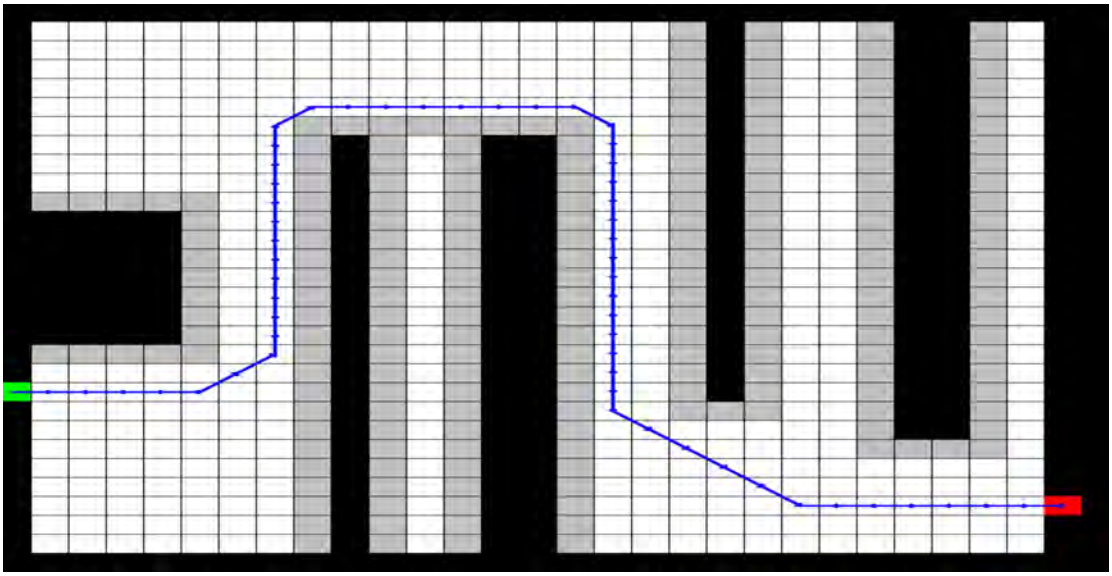


Figure 5.1 Path planning grid map.

After that, neighboring solutions of the current solution are generated in order to explore the search space of the problem. So that, four methods (*elitism*, *random*, *crossover* and *mutation*) were implemented to generate the solutions, and a random operator is used for each iteration.

The random choice of the used operator gives the algorithm both the explorative and exploitative features, that are useful in escaping local minimum, and finding better solution through searching in the neighbors of elite solutions respectively.

In order to start the SA algorithm, the cooling schedule is defined, which consists of four variables (initial temperature, final temperature, annealing schedule and number of iterations per temperature).

The **initial temperature** is the starting temperature of the algorithm, and the **final temperature** is the temperature, at which the algorithm becomes a greedy algorithm, and does not accept worse solutions. The **annealing schedule** is the variable that decides how the temperature is decremented from the initial temperature to the final temperature, throughout the algorithm iterations, based on [141] the geometric cooling schedule is used where the current temperature $T(t)$ is calculated as shown in Equation 5.1:

$$T(t) = T_o \alpha^i, \quad where \quad i = 1, 2, ..., i_f \qquad \& \quad 0 < \alpha < 1 \tag{5.1}$$

where T_o is the initial temperature that is set to 5000, α is the geometric cooling coefficient that is set to 85%, and i is the number of iteration.

The **number of iterations per temperature** is responsible of stabilizing the system at this temperature, through several exploration of the search space, at approximately same acceptance ratio of worst solutions. The acceptance ratio is calculated as show in Equation 5.2, at which the condition to accept is to have the estimated probability coefficient greater than 85%.

$$p = exp^{\frac{-(NL-CL)}{T}} \tag{5.2}$$

where p is the probabilistic coefficient, NL is the neighbor solution length, and CL is the current solution length.

The optimization process aims to find the optimal solution to a certain problem by maximizing or minimizing a certain function. This function is called the objective function of the optimization problem, which measures the quality of the provided solution. This objective was explicitly taken into consideration during the formulation of the path planning problem as the minimization of the distances between the waypoints, from the start point to the destination point, as shown in Equation 5.3.

$$L = min\left(\sum d_{i,j}\right) \tag{5.3}$$

where L is the total length of the solution and $d_{i,j}$ is the distance between each two waypoints.

In order not to stuck in a local minimum, the SA algorithm combines greedy strategy to search for better solutions and stochastic strategy. Therefore, the neighboring solution is directly accepted, ***if and only if*** its quality is better than the previous solution. The algorithm results a vector of waypoints in terms of X and Y coordinates, taking into consideration a safety distance of 1 m from the occupied cells. This is mainly to avoid the air resistance effect from the walls on the quadcopter, which affects its balance.

5.1.2 Fuzzy Logic Control

Fuzzy Logic Control (FLC) is considered as one of the smart control systems, which based on the concept of expressing the control problem with natural language (set of linguistic rules), based on logical statements (***if, then***) [343]. FLC has the ad-

vantage over the genetic algorithms and the neural networks, because it provides solutions that can be understandable by the human operator [117]. As it shown in Figure 5.2, FLC consists of four main parts:

- **Fuzzification:** is the process of transforming the crisp inputs of the control system, using the linguistic expression, so that it can be compared to the rules in the rule-base.
- **Rule-Base:** or the knowledge of the fuzzy system, which contains a set of ***if, then*** rules that describe the control laws, and governs the control actions.
- **Inference Engine:** is the mechanism that evaluates the suitable rule and selects the relevant input to the system. Inference engine has two main strategies: **Implication:** which is in charge of evaluating the control actions individually based on the rules and the input values. **Aggregation:** evaluates the set of control actions, by adding all control actions from all the rules of the rule base in a manner that is predefined by the operator.
- **Defuzzification:** transforms the fuzzy results obtained from the inference engine to the crisp outputs, that can be as inputs to the system.

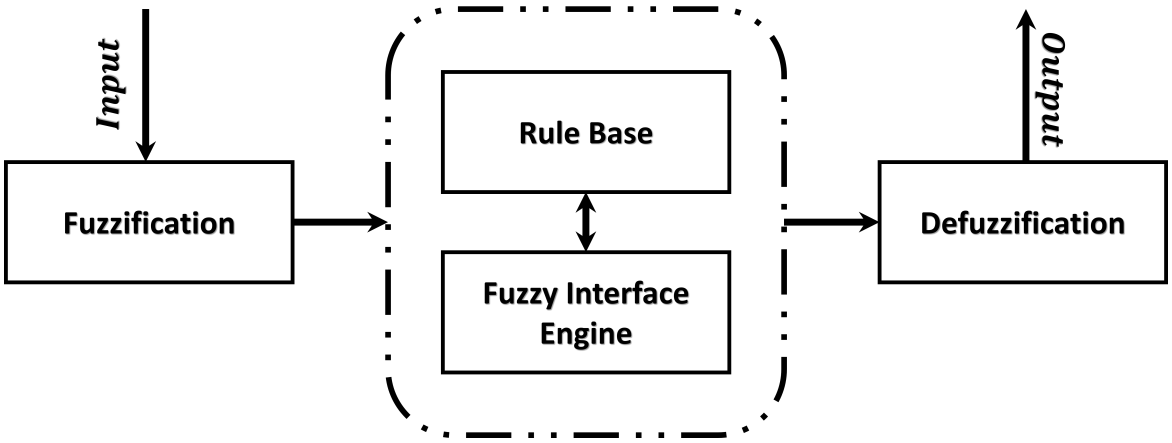
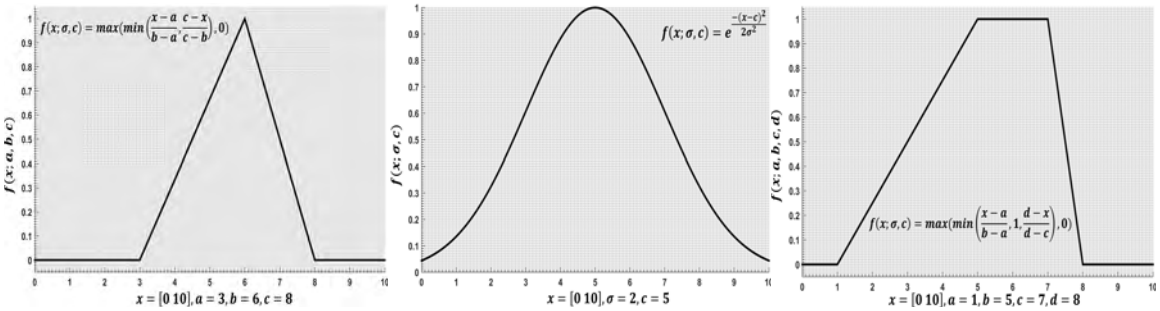


Figure 5.2 Fuzzy Logic Control.

In addition, FLC has three main parameters:

- **Membership Functions:** is the graphical representation of the fuzzy set of rules. There are three different types of membership functions: Triangular (Figure 5.3a), Gaussian (Figure 5.3b), and Trapezoidal (Figure 5.3c).

- **Rule Weight:** the FLC uses this parameter in order to scale the membership functions of the output linguistic variables.



(a) Triangular (b) Gaussian (c) Trapezoidal

Figure 5.3 Types of membership functions

The methodology of the presented Fuzzy Control in the proposed UAV navigation system is based on the method of Mamdani [205–207], where it is reasoned from digital inputs in order to obtain digital outputs by means of the control decision table.

The numerical results are obtained by following the Mamdani method, and it is used to control a non-linear variable; for example, the *roll* or *pitch* angle of a UAV. Thus, Mamdani method allows a numerical result, where the numeric variables are interpreted by fuzzy sets, and following a group of rules related to the fuzzy sets (input and output), thereafter, specific numeric values are assigned to the output Fuzzy set.

The numeric values of the output variable are obtained through the global mass center, in order to find the crisp single output value in the defuzzication step. The weight of the output is estimated by using the following formula:

$$z_{out} = \frac{\sum_{i=1}^{n} z_i \prod \mu(z_i))}{\sum_{i=1}^{n} \prod \mu(z_i))} \tag{5.4}$$

This methodology is applied in order to control the stability, as well as the positioning for the navigation of the UAV; using numeric information gathered by the on-board sensors. That is, the developed fuzzy controllers maintain the stabilization and the position of the UAV, using the on-board IMU and the monocular camera. The numeric values of the sensors are interpreted by the fuzzy sets; such as **Negative**, **Zero**, **Positive**, which are based on different membership functions; such as, Trapezoidal or Gaussian.

Moreover, in this control architecture, the Mamdani-based fuzzy controller uses the *(x, y, z)* position estimated by the vision system, in order to generate the appropriate control commands, and perform the correct maneuver to navigate autonomously.

As it is illustrated in Figure 5.4, the overall controllers have been developed in order to control the motion of the UAV; by tracking and following predefined waypoints, as well as to keep the stability of the flight. The notations that are used in the rule base to represent the input and output values are defined in Table 5.1.

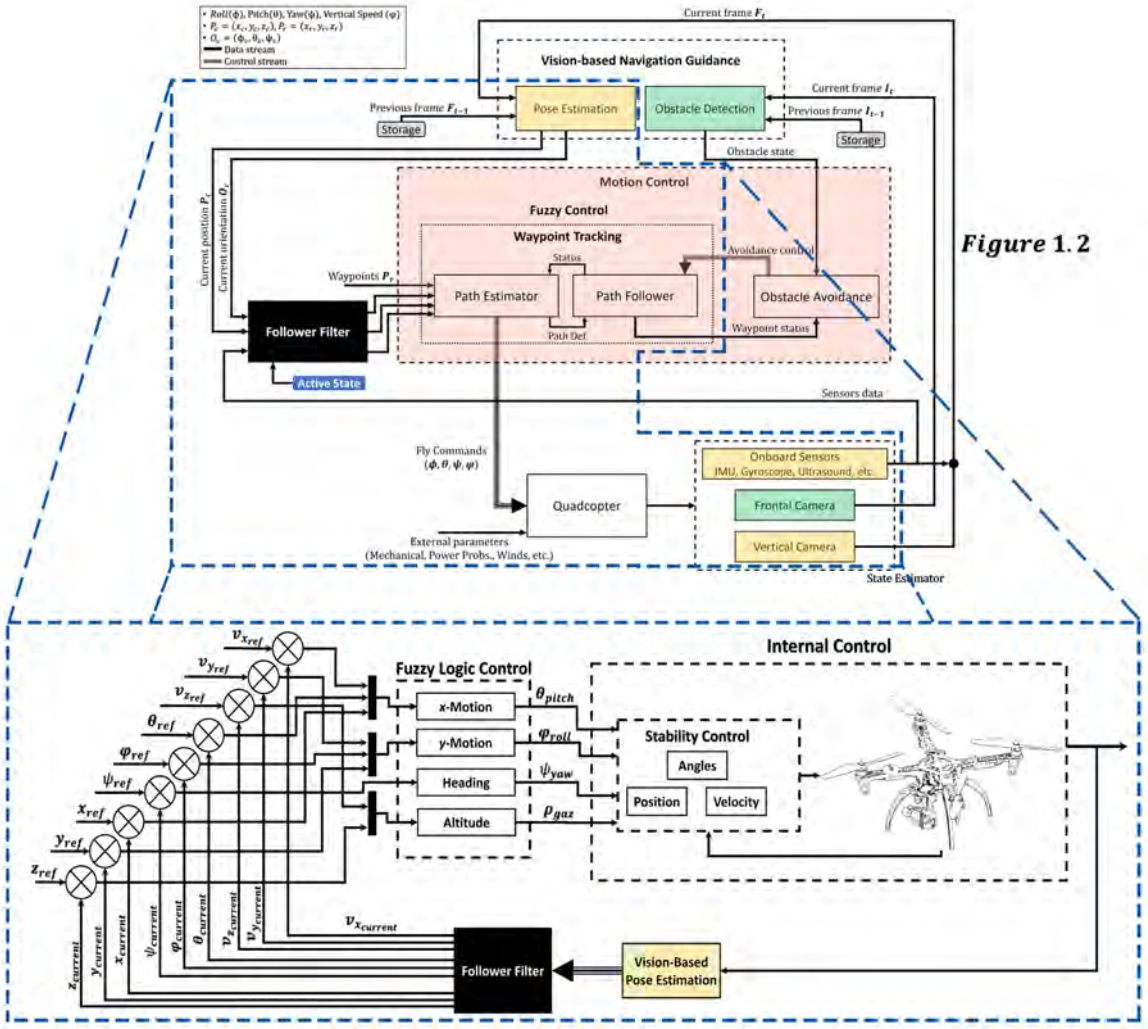


Figure 5.4 Waypoint Tracking Control System.

The fuzzy controller of both stability and position modes is explained in the following:

Table 5.1 Fuzzy rule base notation

Notation	Representation
NB	Negative-Big
NM	Negative-Medium
N	Negative
Z	Zero
P	Positive
PM	Positive-Medium
PB	Positive-Big

Altitude and Yaw Controllers

The controller of the altitude and the heading angle (*yaw*) has been defined in a simple way, with one input and one output. Tables 5.2 and 5.3 show the direct rules between the inputs (*altitude* and *yaw*), and the outputs (*control command value*).

Table 5.2 *Altitude* control decision table

	Altitude (mm)						
	NB	**NM**	**N**	**Z**	**P**	**PM**	**PB**
Altitude Output Control	NB	NM	N	Z	P	PM	PB

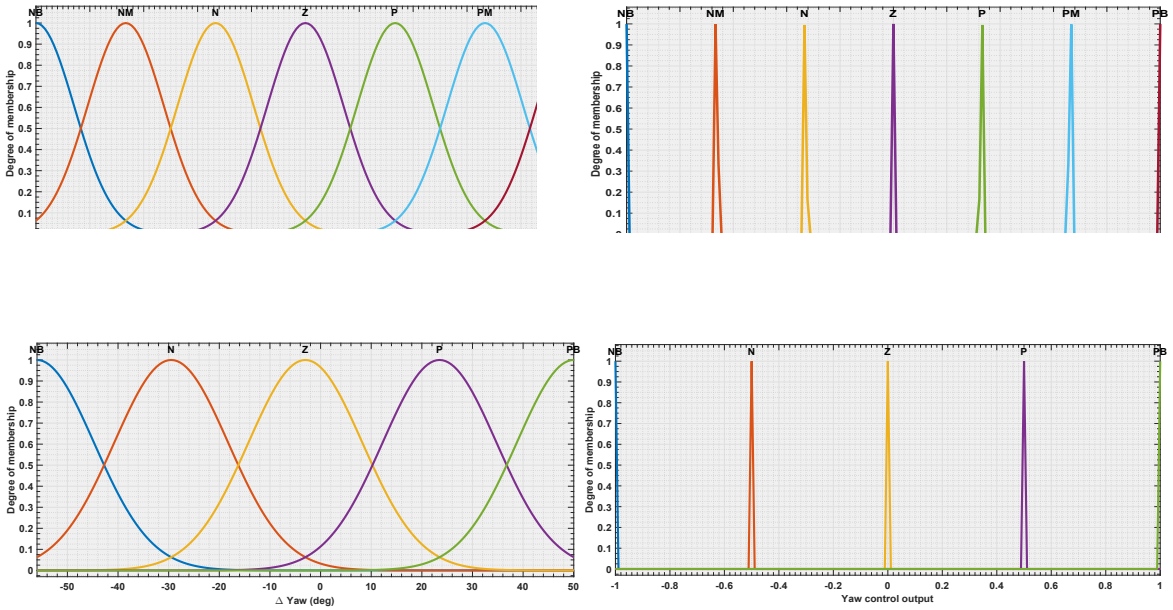
The inputs *altitude* and *yaw* in the base of rules are defined as the error between the reference and current values as follows:

$$\begin{aligned} altitude &= altitude_{ref} - altitude_{current} \\ yaw &= yaw_{ref} - yaw_{current} \end{aligned} \tag{5.5}$$

Table 5.3 *Yaw* control decision table

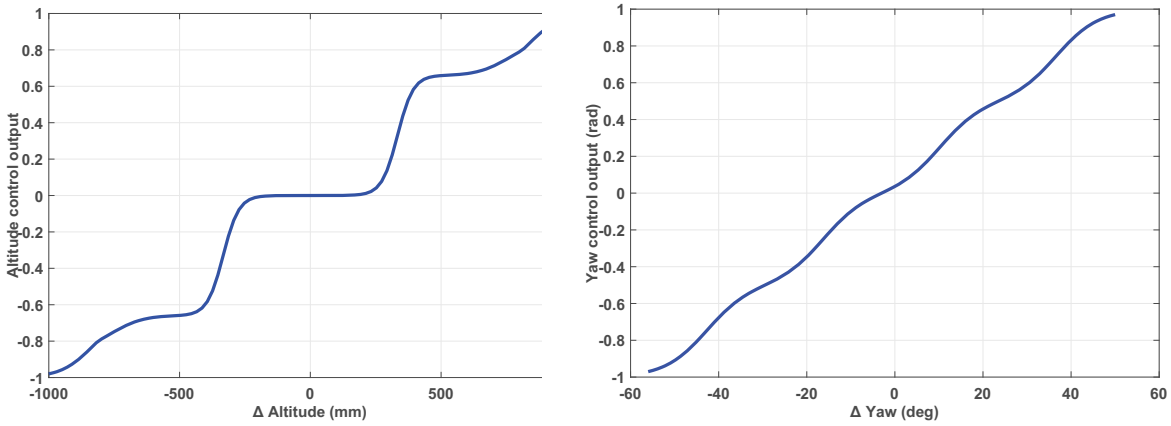
	Yaw Angle (deg)				
	NB	**N**	**Z**	**P**	**PB**
Yaw Output Control	NB	N	Z	P	PB

As it is shown in Figure 5.5, the changes in the inputs and the output control of the altitude and the heading angle is based on the Gaussian membership functions. Furthermore, Figure 5.6 shows the control surface of these properties.



(c) Δ Yaw input (d) Yaw control output

Figure 5.5 Membership functions of *Altitude* and *Yaw* controllers; **a:** Δ Altitude input, **b:** Altitude control output, **c:** Yaw input, and **d:** Yaw control output.

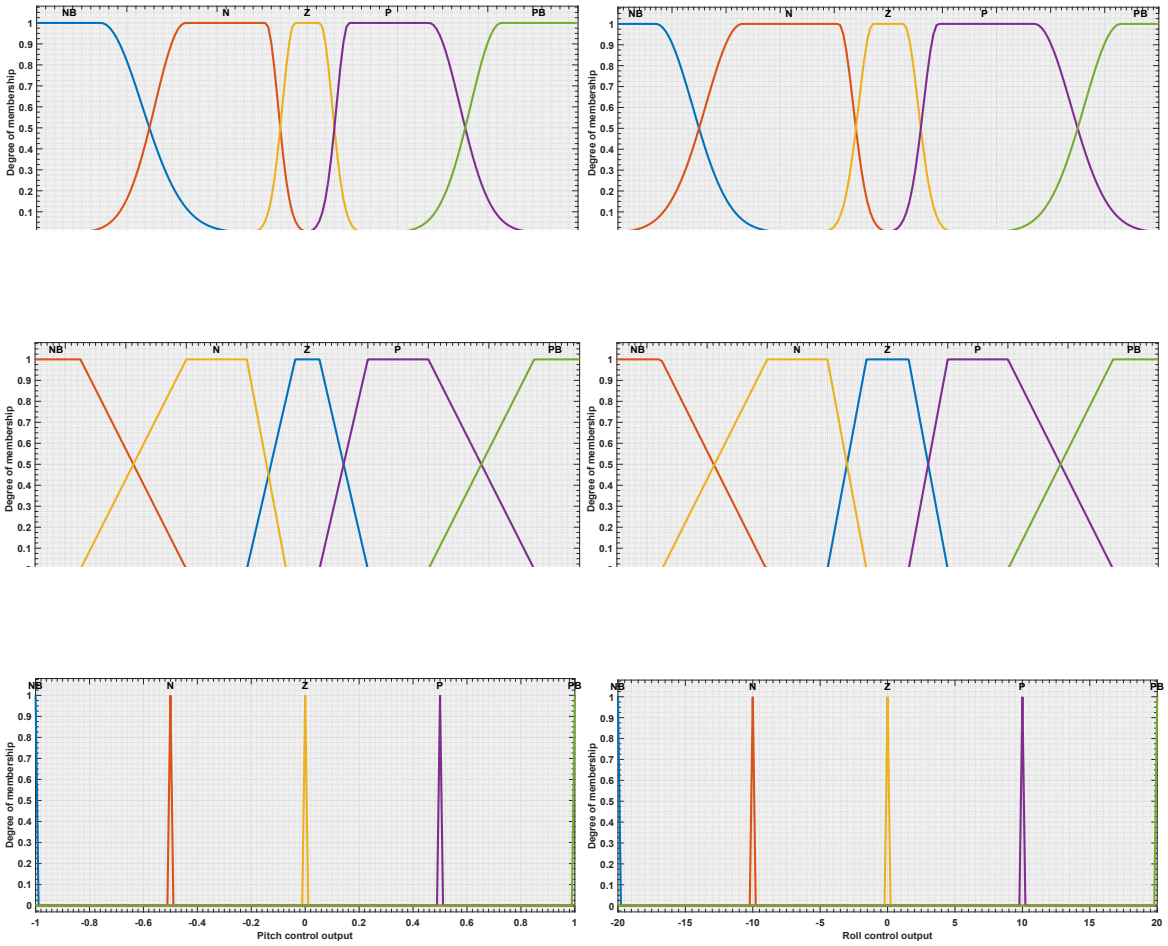


(a) *Altitude* Control Surface. (b) *Yaw* Control Surface.

Figure 5.6 Control Surfaces; **a:** *Altitude*, and **b:** *Yaw*.

Stability Fuzzy Controller

This section presents the controller of the stability mode, at which, the *pitch* and *roll* angles are controlled in order to maintain the UAV flying in a hover mode, with zero *x*-*y* displacement. In this controller, two rule sets have been created (one for each angle). Both controllers achieve the stability of the UAV keeping the displacement

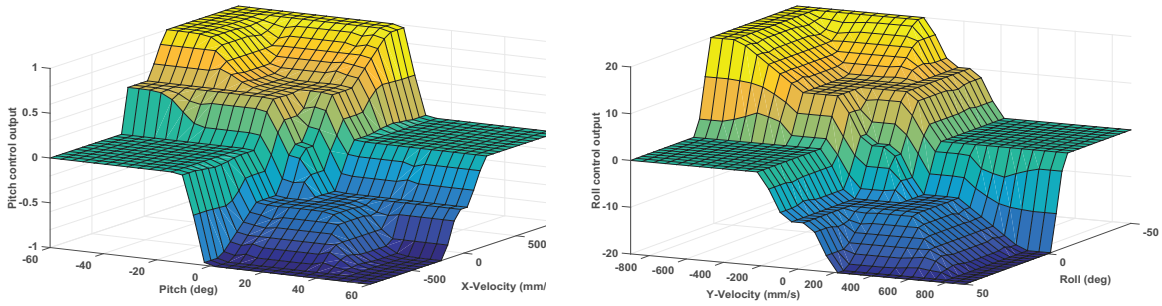


(e) Pitch control output (f) Roll control output

Figure 5.7 Membership functions of *Pitch* and *Roll* Control in Stability mode; **a:** Δ *Pitch* input, **b:** Δ *Roll* input, **c:** Δ x-velocity input, **d:** Δ y-velocity input, **e:** Pitch control output, and **f:** Roll control output.

The angles values (*roll* and *pitch*) are automatically adjusted, in order to maintain the position of the UAV with horizontal velocities V_x and V_y equal to zero. The angles and the velocities are estimated from the on-board IMU without considering any positioning information from the vision system. Thus, for each angle output control, two inputs are defined (horizontal velocity and its corresponding angle).

The controller is developed based on five sets for each input, using trapezoidal memberships functions. In which, it has been found that using the trapezoidal functions provides more stability in the motion of controlling the horizontal velocities and (roll, pitch) angles than the Gaussian functions, as it is shown in Figure 5.7.



(a) *Pitch* Control Surface (*Stability*). (b) *Roll* Control Surface (*Stability*).

Figure 5.8 Control Surfaces (Stability mode); **a:** Pitch, and **b:** Roll.

In addition, Figure 5.8 shows the control surface of the two angles (*pitch* and *roll*).

Table 5.4 *Pitch* control decision table (***Stability mode***)

X-Velocity (mm/s) \ **Pitch Angle** (deg)	**NB**	**N**	**Z**	**P**	**PB**
NB	Z	Z	NB	NB	NB
N	Z	Z	N	N	NB
Z	P	P	Z	N	N
P	PB	P	P	Z	Z
PB	PB	PB	PB	Z	Z

The controller is achieved by the fuzzy rule-base matrix, which is composed of two decision tables of 25 rules for each angle (*pitch* and *roll*). The rule set of each angle is shown in Tables 5.4 and 5.5, where, the rules are different for each angle, and this is because of their motion behavior; if the *pitch* angle is positive, then V_x is negative, and vice versa, while if the *roll* angle is positive, then the V_y is positive, and vice versa.

Table 5.5 *Roll* control decision table (***Stability mode***)

Y-Velocity (mm/s) \ **Roll Angle** (deg)	**NB**	**N**	**Z**	**P**	**PB**
NB	PB	PB	PB	Z	Z
N	PB	P	P	Z	Z
Z	P	P	Z	N	N
P	Z	Z	N	N	NB
PB	Z	Z	NB	NB	NB

Position Fuzzy-Controller

The third controller is the waypoint tracking mode. In which, the control is designed in order to maintain the position of the UAV (*Hover mode*) or to track and follow the position of the predefined waypoints.

Similarly, to the stability mode, the controller has the angles and the horizontal velocities from the IMU sensor as inputs. However, in this case, the changes in the distance have been added as a new input to the controller.

The distance has been calculated as the difference between the actual UAV position and the hovering point position (*Hover mode*), or the difference of the waypoint position and the UAV current position (*Waypoint tracking*).

Table 5.6 *Pitch* control decision table (***Position mode***)

X-Velocity (mm/s) \ **Pitch Angle** (deg)	$\Delta X < 0$					$\Delta X = 0$					$\Delta X > 0$				
	NB	**N**	**Z**	**P**	**PB**	**NB**	**N**	**Z**	**P**	**PB**	**NB**	**N**	**Z**	**P**	**PB**
NB	Z	Z	N	N	NB	Z	Z	NB	NB	NB	Z	Z	NB	NB	NB
N	P	P	Z	N	N	P	Z	N	NB	NB	P	Z	NB	NB	NB
Z	P	P	P	Z	N	PB	P	Z	N	NB	P	Z	N	N	N
P	PB	PB	P	Z	N	PB	PB	P	Z	N	P	P	Z	N	N
PB	PB	PB	PB	Z	Z	PB	PB	PB	Z	Z	PB	P	P	Z	Z

ΔX (***mm***)

Table 5.7 *Roll* control decision table (***Position mode***)

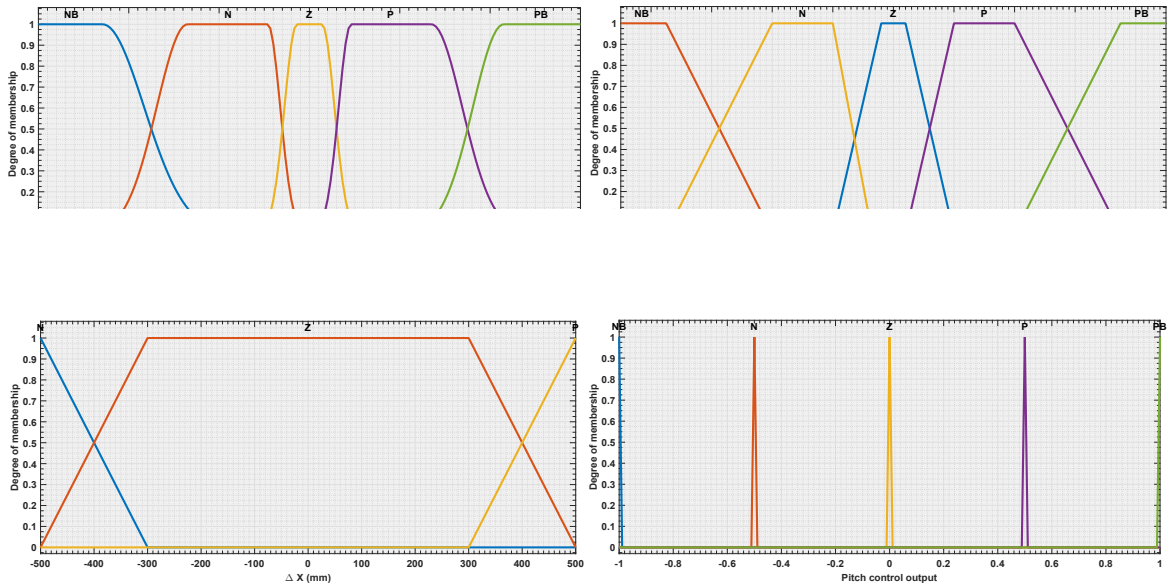
Y-Velocity (mm/s) \ **Roll Angle** (deg)	$\Delta Y < 0$					$\Delta Y = 0$					$\Delta Y > 0$				
	NB	**N**	**Z**	**P**	**PB**	**NB**	**N**	**Z**	**P**	**PB**	**NB**	**N**	**Z**	**P**	**PB**
NB	PB	P	P	Z	Z	PB	PB	PB	Z	Z	PB	PB	PB	Z	Z
N	P	P	Z	N	N	PB	PB	P	Z	N	PB	PB	P	Z	N
Z	P	Z	N	N	N	PB	P	Z	N	NB	P	P	P	Z	N
P	P	Z	N	NB	NB	Z	Z	N	NB	NB	P	P	Z	N	N
PB	Z	Z	NB	NB	NB	P	Z	NB	NB	NB	Z	Z	N	N	NB

ΔY (***mm***)

The new x-y position of the UAV displacement is estimated using the vision-based algorithm that was explained in Chapter 3.

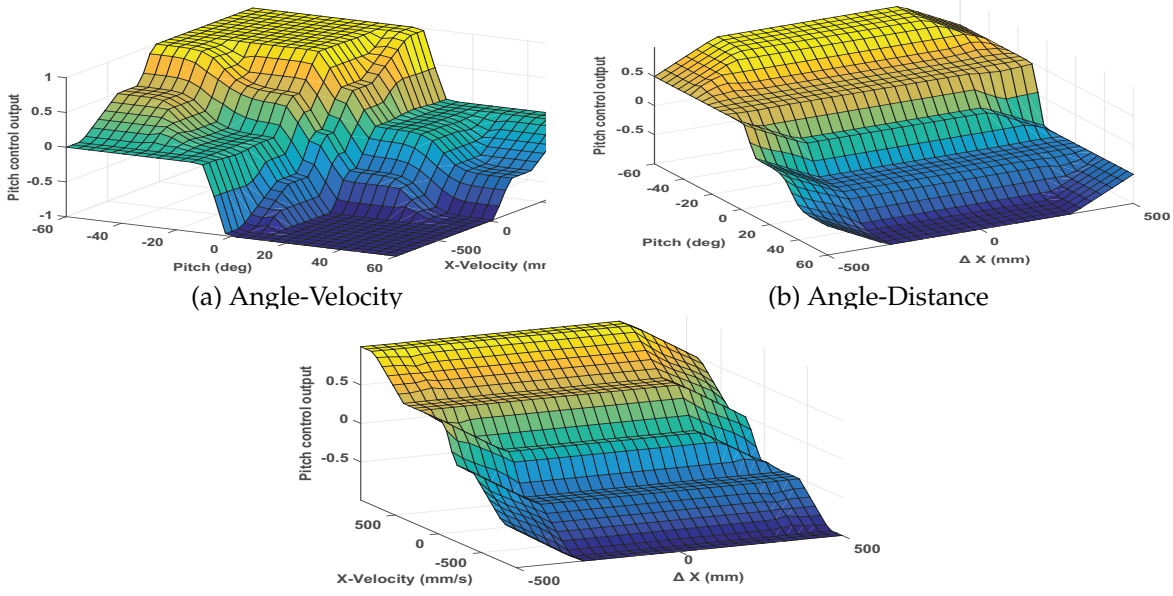
The fuzzy rule-base matrix of each angle (*pitch* and *roll*) of the controller consists of 75 rules, based on the three inputs and one output as it is shown in Tables 5.6 and 5.7, and similarly to the stability mode, Figures 5.9 and 5.11 illustrate the

trapezoidal membership functions of the input and output fuzzy sets, and Figures



(c) Δ *x*-distance input (d) Pitch control output

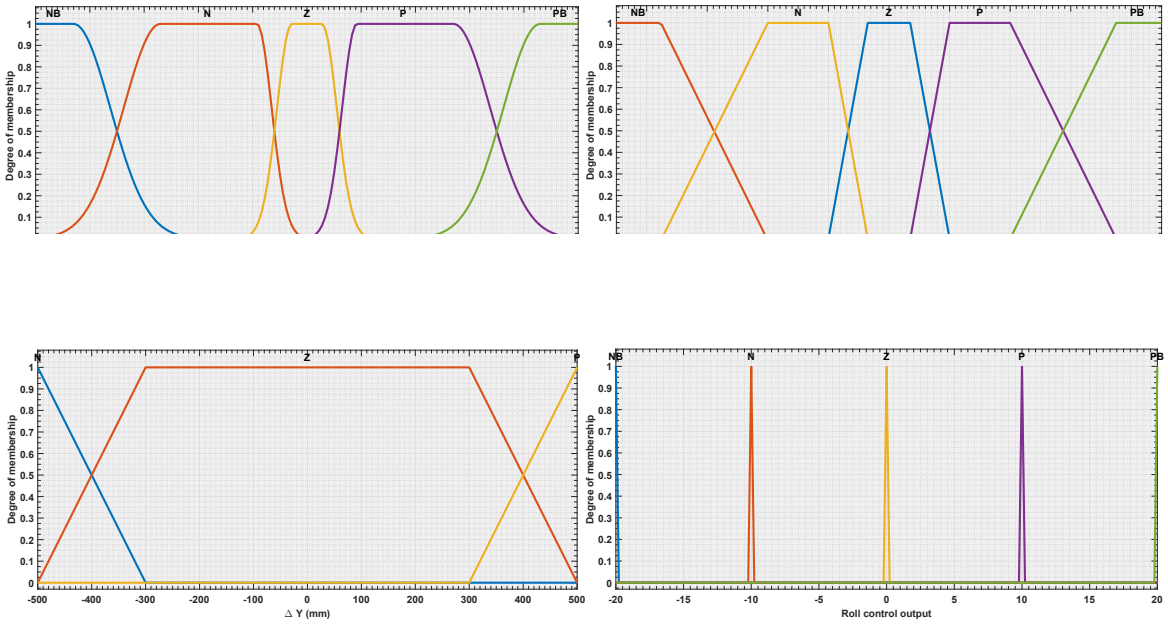
Figure 5.9 Membership functions of *Pitch* Control in Position mode; **a:** Δ *Pitch* input, **b:** Δ x-velocity input, **c:** Δ x-distance input, and **d:** Pitch control output.



(a) Angle-Velocity (b) Angle-Distance

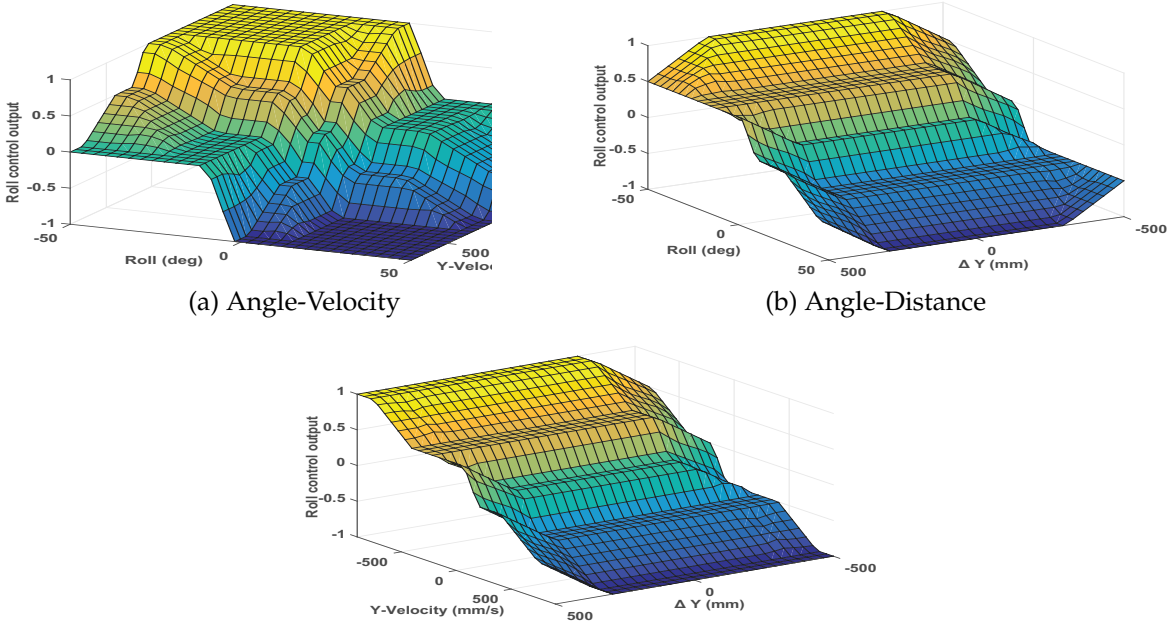
(c) Velocity-Distance

Figure 5.10 Pitch Control Surfaces (Position mode); **a:** Angle-Velocity, **b:** Angle-Distance, and **c:** Velocity-distance.



(c) Δ *y*-distance input (d) Roll control output

Figure 5.11 Membership functions of *Roll* Control in Position mode; **a:** Δ *Roll* input, **b:** Δ y-velocity input, **c:** Δ y-distance input, and **d:** Roll control output.



(a) Angle-Velocity

(b) Angle-Distance

(c) Velocity-Distance

Figure 5.12 Roll Control Surfaces (Position mode); **a:** Angle-Velocity, **b:** Angle-Distance, and **c:** Velocity-distance.

5.1.3 Waypoint Tracking

Flight guidance is the term that defines the ability to follow the predefined path, by tracking the generated waypoints that build this path (waypoint tracking).

Algorithm 5 and Figure 5.13 describe the steps of following and swapping the waypoints in the tracking phase. At which, the algorithm requires the global pose vector of the reference points $(x_r, y_r, z_r, \psi_r)_{1:N}$, and the global current pose of the UAV (x_d, y_d, z_d, ψ_d) as inputs.

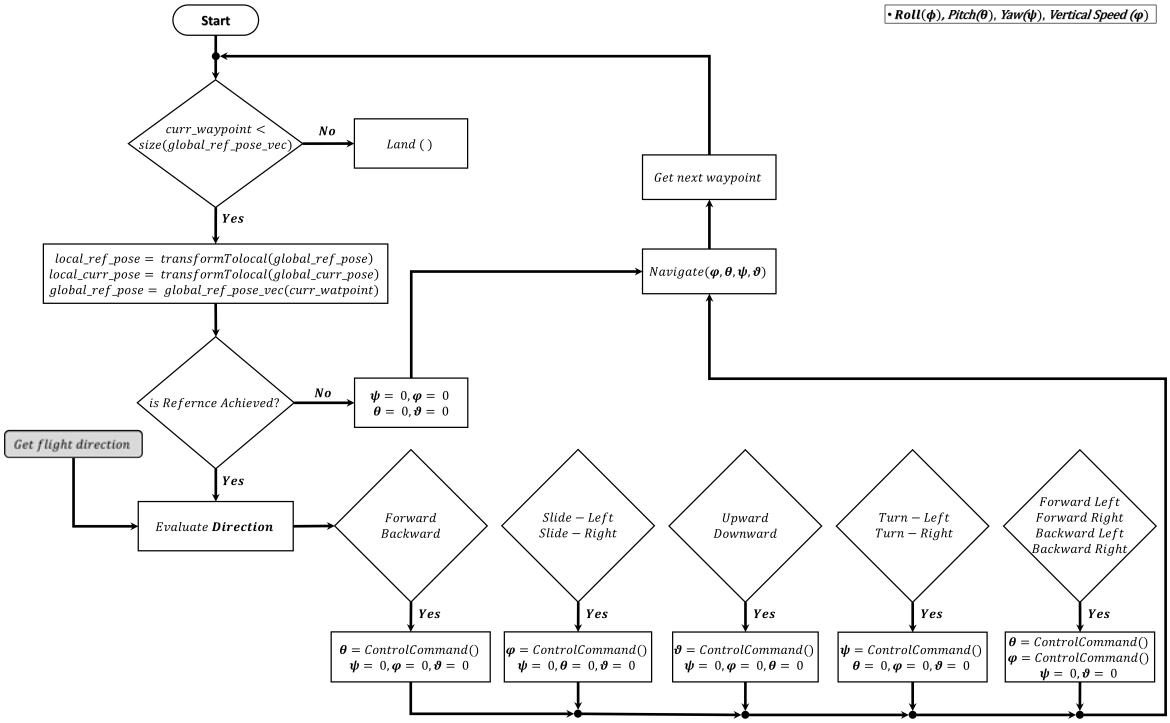


Figure 5.13 Waypoint Tracking.

First, the algorithm reads all the waypoints of the global reference pose vector, and the global current UAV pose as well, and determines the initial heading angle ($\psi_{initial}$), which defines the angle of the UAV in the starting point. Thereafter both poses (*reference* and *current*) are converted to the local coordinates system using the following equations:

$$\begin{aligned} x_{local_t} &= (x_{global_t} - x_{global_{t\text{-}1}})cos(\psi) - (y_{global_t} - y_{global_{t\text{-}1}})sin(\psi) \\ y_{local_t} &= (x_{global_t} - x_{global_{t\text{-}1}})sin(\psi) + (y_{global_t} - y_{global_{t\text{-}1}})cos(\psi) \\ \psi_{local_t} &= \psi_{global_t} - \psi_{initial} \end{aligned} \tag{5.6}$$

where, $(x,y)_{local_t}$ is the current local position of the UAV, $(x,y)_{global_t}$ and $(x,y)_{global_{t-1}}$ are the current and previous global UAV positions respectively, ψ is the heading angle which defines the rotation between the local and the global coordinates, where $\psi = \psi_{global_{initial}} - \psi_{local_{initial}}$, and ψ_{local_t} is the current local heading angle.

The next step is to check if the current reference waypoint has been achieved or not. As it is shown in Figure 5.14, this is estimated by comparing the current UAV pose, and the reference waypoint with the tolerances, which are $\pm 10\%$ of the value of the angle and an area with a radius of 25 cm around the waypoint, where:

$$\begin{aligned} x_{uav_t} &= x_{waypoint_t} \pm 25cm \\ y_{uav_t} &= x_{waypoint_t} \pm 25cm \\ \psi_{uav_t} &= \psi_{waypoint_t} \pm 0.1 \times \psi_{waypoint_t} \end{aligned} \tag{5.7}$$

In the case if the UAV does not achieve the reference waypoint, the algorithm estimates the status and the navigation direction of the UAV, as it is described in Algorithm 6 and shown in Table 5.8. In which, the UAV can perform thirteen possible maneuvers with different navigation directions.

Table 5.8 Possible Navigation Directions

Direction	**Angle**	**Enumerate**
Forward	pitch -	F
Backward	pitch +	B
Slide-Left	roll -	SL
Slide-Right	roll +	SR
Turn-Left	yaw -	TL
Turn-Right	yaw +	TR
Forward Left	pitch - roll -	FL
Forward Right	pitch - roll +	FR
Backward Left	pitch + roll -	BL
Backward Right	pitch + roll +	BR
Hover	pitch = 0 roll = 0 yaw = 0 throttle = 0	H
Upward	throttle +	U
Downward	throttle -	D

Algorithm 5: Waypoints Follower

Input: Global Reference Pose Vector *global_ref_pose_vec*, Global Current Pose *global_curr_pose*
Output: Fly command $Nav(\phi, \theta, \psi, \vartheta)$

Define: Global Reference Pose *global_ref_pose*, Local Reference Pose *local_ref_pose*, Local Current Pose *local_curr_pose*, Current Waypoint *curr_waypoint*, Tolerance *toler*, Direction Enum *nav_dir_enum*, Position Error *pos_err*, Orientation Error *orient_err*
while *curr_waypoint* $<$ *size(global_ref_pose_vec)* **do**
    global_ref_pose $\leftarrow$ *global_ref_pose_vec*(*curr_waypoint*)
    local_curr_pose $\leftarrow$ transformTolocal(*global_curr_pose*)
    local_ref_pose $\leftarrow$ transformTolocal(*global_ref_pose*)
    if !*isRefernceAchieved (local_ref_pose, local_curr_pose, toler)* **then**
        [*nav_dir_enum*, *pos_err*, *orient_err*] $\leftarrow$ getNavigationDirection(*local_ref_pose*, *local_curr_pose*, *toler*) `// See Algorithm 6`
        switch *nav_dir_enum* **do**
            In Parallel
                Process *1*
                    case $'F'$ **do**
                    case $'B'$ **do**
                        $\theta \leftarrow$ ControlCommand(*pos_err*), $\phi \leftarrow 0$, $\psi \leftarrow 0$, $\vartheta \leftarrow 0$
                    case $'SL'$ **do**
                    case $'SR'$ **do**
                        $\phi \leftarrow$ ControlCommand(*pos_err*), $\theta \leftarrow 0$, $\psi \leftarrow 0$, $\vartheta \leftarrow 0$
                    case $'FL'$ **do**
                    case $'FR'$ **do**
                    case $'BL'$ **do**
                    case $'BR'$ **do**
                        $\phi \leftarrow$ ControlCommand(*orient_err*),
                        $\theta \leftarrow$ ControlCommand(*orient_err*), $\psi \leftarrow 0$, $\vartheta \leftarrow 0$
                Process *2*
                    case $'U'$ **do**
                    case $'D'$ **do**
                        $\vartheta \leftarrow$ ControlCommand(*pos_err*), $\phi \leftarrow 0$, $\theta \leftarrow 0$, $\psi \leftarrow 0$
                    case $'TL'$ **do**
                    case $'TR'$ **do**
                        $\psi \leftarrow$ ControlCommand(*orient_err*), $\phi \leftarrow 0$, $\theta \leftarrow 0$, $\vartheta \leftarrow 0$
    else
        $\psi \leftarrow 0$, $\phi \leftarrow 0$, $\theta \leftarrow 0$, $\vartheta \leftarrow 0$
    $Nav(\phi, \theta, \psi, \vartheta)$
    curr_waypoint $\leftarrow$ *curr_waypoint* $+ 1$
$Land()$

Algorithm 6: Get Navigation Direction

Input: Reference Pose ref_pose, Current Pose $curr\_pose$, Tolerance $toler$
Output: Navigation Direction Enum nav_dir_enum, Position Error pos_err, Orientation Error $orient\_err$

$pos\_err.x \leftarrow (ref\_pose.x \pm toler) - curr\_pose.x$,
$pos\_err.y \leftarrow (ref\_pose.y \pm toler) - curr\_pose.y$,
$pos\_err.z \leftarrow (ref\_pose.z \pm toler) - curr\_pose.z$
$orient\_err \leftarrow (ref\_pose.\psi \pm toler) - curr\_pose.\psi$
In Parallel
  Process *1*
    else if $pos\_err.x > 0$ *and* $pos\_err.y = 0$ **then**
      $nav\_dir\_enum \leftarrow' F'$
    else if $pos\_err.x > 0$ *and* $pos\_err.y > 0$ **then**
      $nav\_dir\_enum \leftarrow' FR'$
    else if $pos\_err.x > 0$ *and* $pos\_err.y < 0$ **then**
      $nav\_dir\_enum \leftarrow' FL'$
    else if $pos\_err.x < 0$ *and* $pos\_err.y = 0$ **then**
      $nav\_dir\_enum \leftarrow' B'$
    else if $pos\_err.x < 0$ *and* $pos\_err.y > 0$ **then**
      $nav\_dir\_enum \leftarrow' BR'$
    else if $pos\_err.x < 0$ *and* $pos\_err.y < 0$ **then**
      $nav\_dir\_enum \leftarrow' BL'$
    else if $pos\_err.x = 0$ *and* $pos\_err.y > 0$ **then**
      $nav\_dir\_enum \leftarrow' SR'$
    else if $pos\_err.x = 0$ *and* $pos\_err.y < 0$ **then**
      $nav\_dir\_enum \leftarrow' SL'$
  Process *2*
    if $pos\_err.z > 0$ **then**
      $nav\_dir\_enum \leftarrow' U'$
    else if $pos\_err.z < 0$ **then**
      $nav\_dir\_enum \leftarrow' D'$
  Process *3*
    else if $orient\_err > 0$ **then**
      $nav\_dir\_enum \leftarrow' TR'$
    else if $orient\_err < 0$ **then**
      $nav\_dir\_enum \leftarrow' TL'$
return nav_dir_enum, pos_err, $orient\_err$

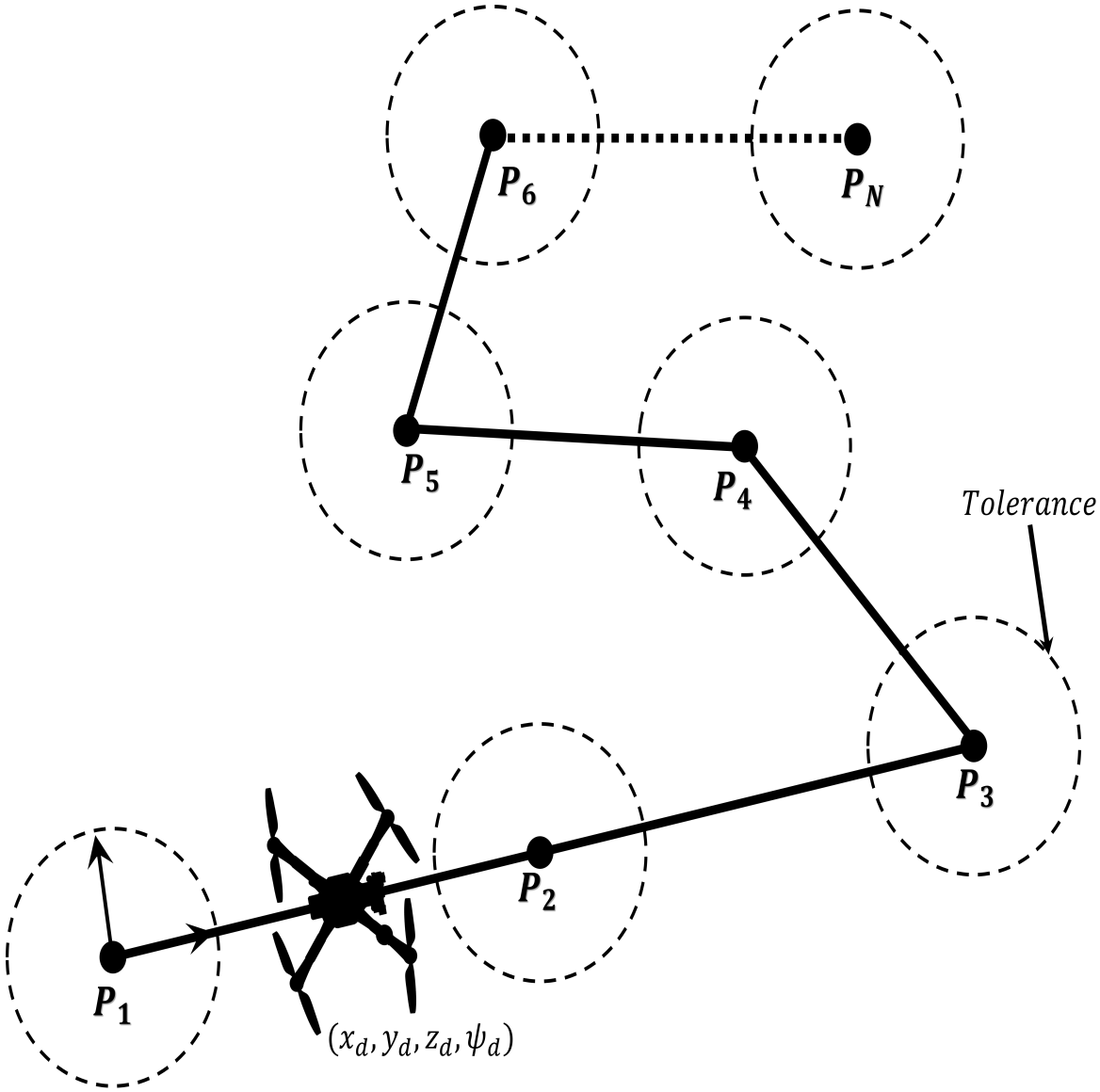


Figure 5.14 Waypoint Tracking (Following and Swapping).

These navigation directions are estimated by comparing the position and the heading angle of the reference waypoint, with the current pose of the UAV, for example; as it is shown in Figure 5.15a, if the position of the reference waypoint in x-axis is greater than the UAV x-position, the reference position is equal to the UAV position in both y and z axes, and the difference in the heading angle $\psi_{refernce} - \psi_{uav} = 0$, then the navigation direction will be Forward with an enumerate **'F'**.

Another example is illustrated in Figure 5.15b, in which, the reference waypoint

position in all the axes is grater than UAV position, and the difference in the heading angle between the reference waypoint and the UAV is negative, thereafter, the UAV performs a Turn-Left **'TL'** and Upward **'U'** maneuvers coupled with the motion in the Forward-Left direction **'FL'**, and so on. The UAV displays this motion behavior, because the (x, y) position control system is applied in parallel on the altitude and the heading angle.

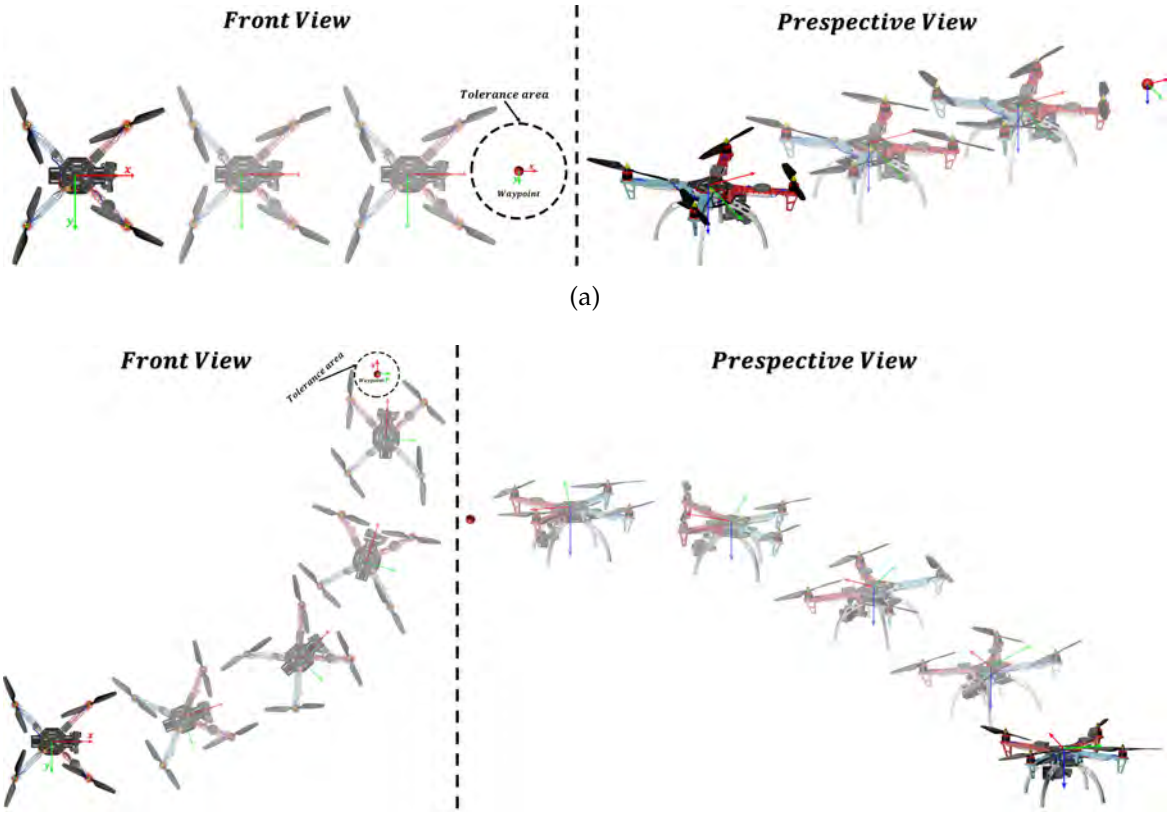


Figure 5.15 Example of UAV motion behavior; **a:** Straight, and **b:** Diagonal.

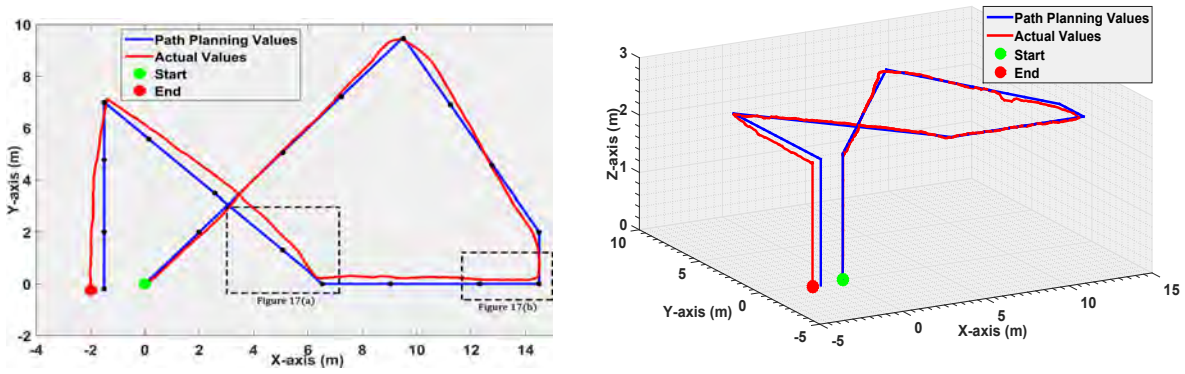
Finally, after estimating the navigation direction of the UAV. The fuzzy controller is applied in order to send the flight commands to the UAV. This process keeps tracking the waypoints till the end of the size of the global pose vector of the reference points. After reaching the last point, the UAV performs an autonomous landing maneuver.

5.2 Results (*Waypoint Tracking*)

Various outdoor and indoor experiments were tested; in order to evaluate and verify the performance of the control system on the UAV, to track the predefined waypoints. The occupancy grid maps for the four scenarios are generated to the scale of the real dimensions, the grid cells are divided into one square meter ($1m^2$). All the static obstacles are considered in the mapping process. Subsequently, the maps are loaded into the program in order to start the path planning algorithm and to obtain the final 3D waypoints in order to achieve the autonomous navigation. In these experiments, the same platforms are used like in Chapter 3.

5.2.1 Outdoor Scenario

In Figure 5.16, it has been shown the performance of the UAV in navigating a random maneuvers of a total distance 53.9m. At which the UAV follows a straight path with different range of *yaw* angles circumnavigating static obstacles, maintaining the safe distance to them.

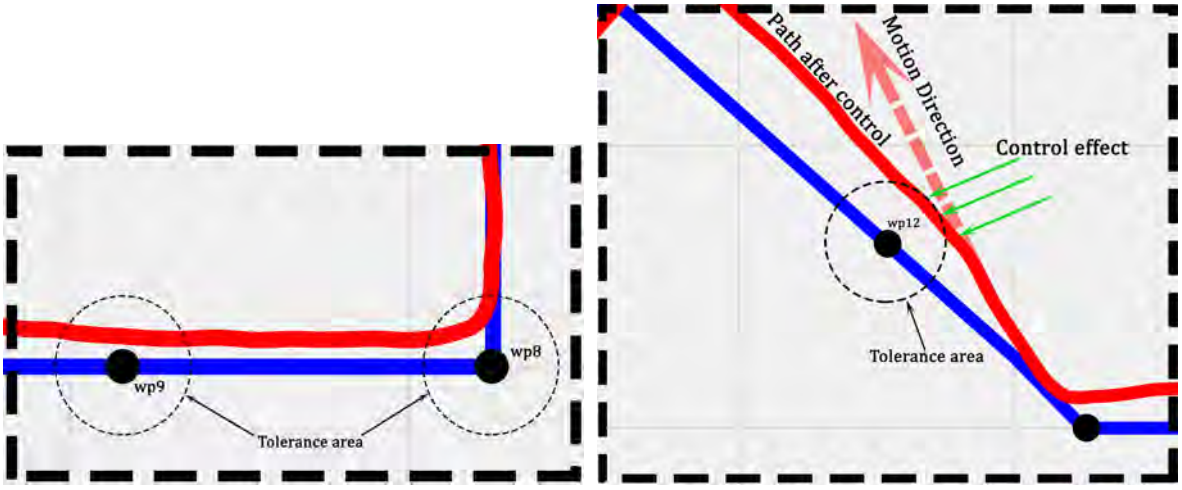


(a) 2D Waypoint Tracking

(b) 3D Waypoint Tracking

Figure 5.16 Waypoint Tracking Diamond.

In Figure 5.17a, the strategy of the waypoint switching is shown in details, at which, once the UAV enters the tolerance area of the waypoint (wp_8), the control system receives the position of the new waypoint (wp_9), and starts changing the direction of the UAV, and performs the maneuver before reaching the exact position of the previous waypoint (wp_8). This case demonstrates the concept of the path-following priority than the waypoint-precision.



(a) (b)

Figure 5.17 Detailed waypoint tracking behaviors (*Diamond*).

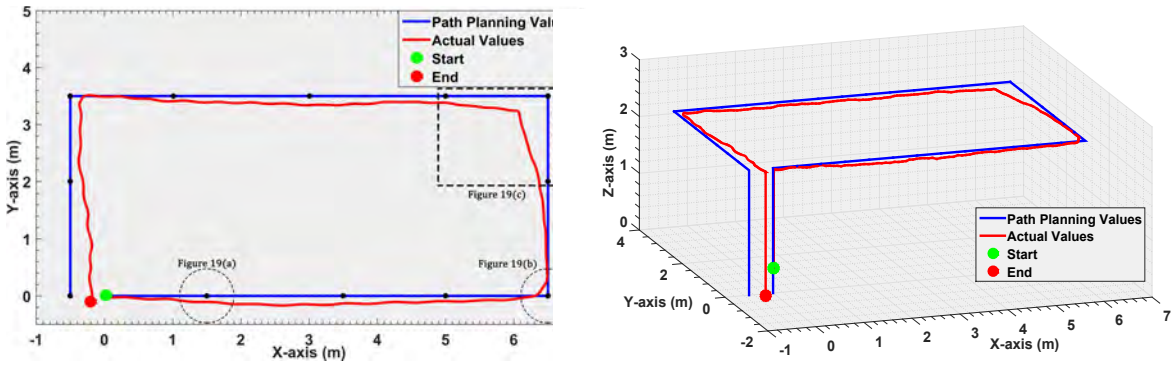
The effect of the control system in order to maintain the UAV to be in the path is shown in Figure 5.17b. At which, once the UAV deviates from its predefined path (outside the tolerance area), the control system instantly corrects the UAV motion in order to be inside the tolerance area of the corresponding waypoint (wp_{12}).

The obtained results showed the high performance of the algorithms in following the theoretical path, with total error in the final point of $\pm$30cm in *(x,y)*-direction (Figure 5.16a) and $\pm$5cm in the *z*-direction(Figure 5.16b).

5.2.2 Indoor Scenario

Similarly to the outdoor scenarios, Figure 5.18 illustrates the performance of the UAV in navigating an open ending square shape with change of 90^o in *yaw* angle and total distance of 20.5m. This experiment shows the accurate performance of the UAV control system to follow a loop shape, where the starting point and final point are close to each other.

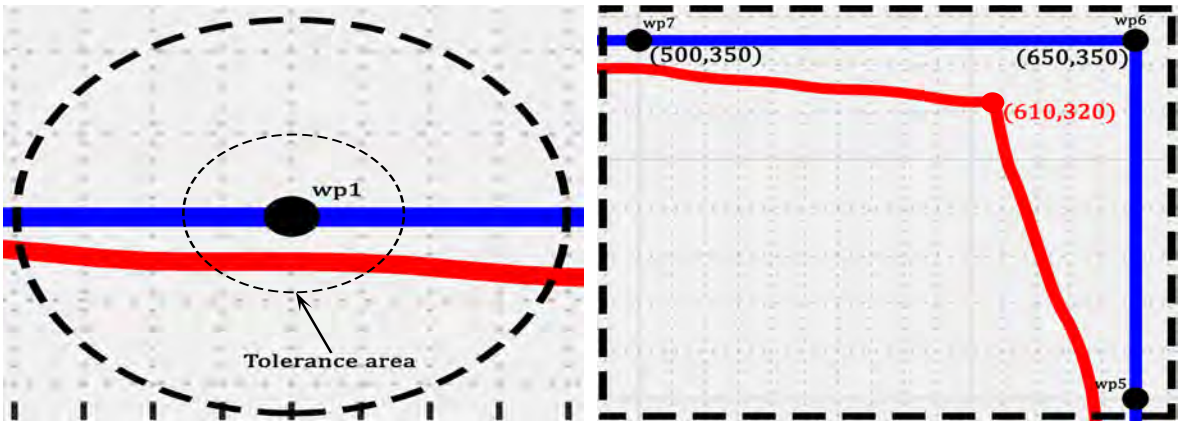
Figure 5.19 shows different situations and behaviors based on the waypoint tracking and the control algorithms. In Figure 5.19a, it is shown that in the case when the UAV flight in a *forward* direction, and its *(x, y)*-position is equal to the waypoint position (wp_1) with the tolerance, the tracking algorithm starts looking for the next waypoint to follow, without the necessity of passing through the exact location of the waypoint.



(a) 2D Waypoint Tracking (b) 3D Waypoint Tracking

Figure 5.18 Waypoint Tracking Square.

The situation of a missed waypoint is shown in Figure 5.19b. In this case the UAV was affected by the vortices that are reflected from the wall close to it. This effect pushes the UAV far from the waypoint. It is shown that the current x-position of the UAV is located beyond the waypoint (wp_6), thereafter the tracking algorithm looks for the next waypoint (wp_7) instead of performing *backward* maneuver.



(a) (b)

Figure 5.19 Detailed waypoint tracking behaviors (*Square*).

Similarly to the outdoor experiments, the obtained results show near optimal navigation with overall error in the final point of ±30cm in *(x,y)*-direction (Figure 5.18a) and ±5cm in the z-direction (Figure 5.18b).

From the experiments, the obtained results illustrate the proposed algorithms prioritize following the path segments smoothly, instead of passing by the exact waypoint location. This is achieved by implementing the early waypoint switching strategy, once the UAV enters the tolerance area around the waypoint.

5.3 Obstacle Avoidance

In this section, the combined mission of the waypoint tracking with the obstacle avoidance method is described. The geometrical problem is illustrated in Figure 5.20 while the avoidance technique is summarized in Algorithm 7.

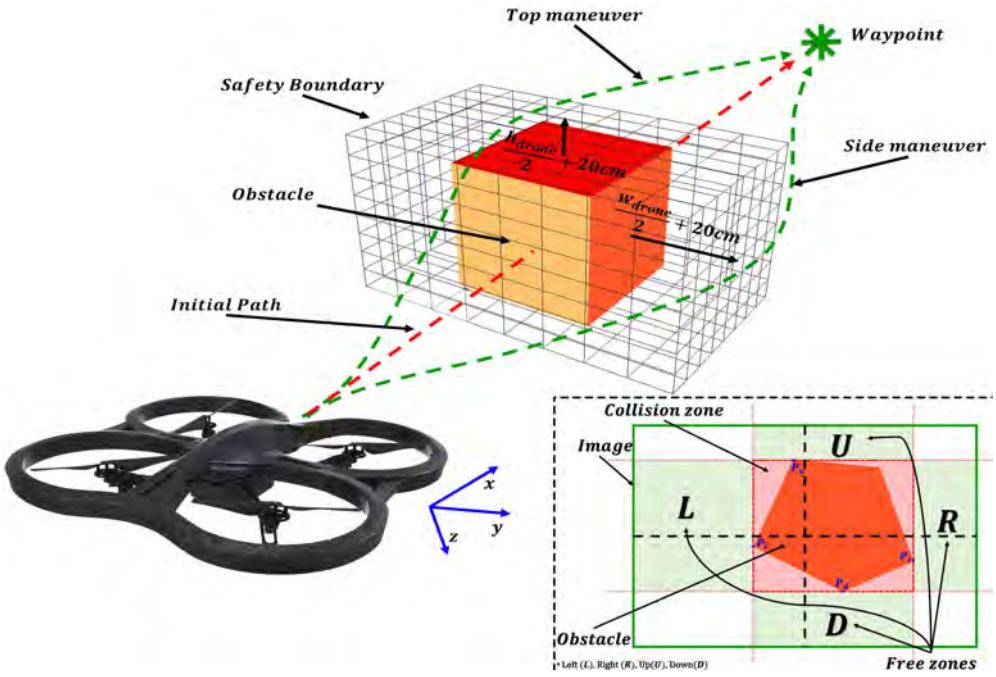


Figure 5.20 Obstacle Avoidance Decision

To define the problem of the waypoint tracking, let the UAV X flying at a velocity V, considering the UAV flies forward at a constant velocity along its *x-axis*, where:

$$\begin{aligned} X &= \begin{bmatrix} x_d & y_d & z_d \end{bmatrix}^T \quad and, \\ V &= \begin{bmatrix} u_d & v_d & w_d \end{bmatrix}^T \end{aligned} \tag{5.8}$$

On the other hand, let a waypoint:

$$WP = \begin{bmatrix} x_w & y_w & z_w \end{bmatrix}^T \tag{5.9}$$

Hence, the waypoint is assumed to be tracked if x_d is achieved when both y_d and z_d are satisfied, where:

$$\begin{aligned} x_d &= x_w \pm \mu_x \\ y_d &= y_w \pm \mu_y \\ z_d &= z_w \pm \mu_z \end{aligned} \tag{5.10}$$

where, μ is the tolerance area around the waypoint position with a radius of 10 cm from the waypoint.

Let a frontal obstacle ***obs*** be detected by Algorithm 3 (Chapter 4), situated in the path of the UAV and surrounded by the collision-free zones $\tau = (\tau_l, \tau_r, \tau_u, \tau_d)$.

First, the avoidance algorithm checks all the collision-free zones and differentiate which zone is the best to be followed. This is achieved by reading the position of the next waypoint and by comparing the size of the free zones, where the final maneuver will be in term of **Left-Right** or **Up-Down** motion, or a combination of both. After that, a safety boundary surrounding the obstacle is assumed as it is illustrated in Figure 5.20, which is based on the dimensions of the UAV. This safety region is estimated to be:

$$\begin{aligned} Safety_{lr} &= \left(\frac{w_{uav}}{2}\right) + 20\ cm \\ Safety_{ud} &= \left(\frac{h_{uav}}{2}\right) + 20\ cm \end{aligned} \tag{5.11}$$

where, w and h defines the width and the height of the UAV respectively.

Afterwards, the algorithm reads the position of the predefined next waypoint, and calculates the new waypoint out of the path (in order to avoid the obstacle), and sends the control command (velocity control) to the UAV for maneuvering according to the waypoint position as follows:

- Horizontal maneuver (*Right* or *Left*)

$$V_{lr} = (\kappa_p \times err_t) + (\kappa_i \times Ierr_t) + (\kappa_d \times \frac{err_t - err_{t-1}}{dt}) \tag{5.12}$$

- Vertical maneuver (*Top* or *bottom*)

$$V_{ud} = (\iota p \times err_t) + (\iota i \times Ierr_t) + (\iota d \times \frac{err_t - err_{t-1}}{dt}) \tag{5.13}$$

  where y_d and z_d are the UAV position in (y, z)-coordinates, κ, ι are the control coefficients, $Ierr$ is the accumulated integral error, where $Ierr = \sum err \times dt$, and err is the feedback error which is $err = y_d \pm Safety_{lr}$ or $err = z_d \pm Safety_{ud}$.

Algorithm 7: Obstacle Avoidance Algorithm

Input: Collision-free zone $(\tau_l, \tau_r, \tau_u, \tau_d)$, UAV position $X = (x_d, y_d, z_d)$
Output: Navigation command $Nav(\phi, \theta, \psi, \vartheta)$
Define: Roll ϕ, Vertical speed ϑ, Next waypoint $nwp = (x_w, y_w, z_w)$ and Maneuver behavior (Mr_{rl}, Mr_{ud})
begin
  while *!atGoal* **do**
    if $ObAreaRatio \leq \underline{2}$ *and* $KPSizeRatio \leq \underline{\underline{1.5}}$ **then**
      $nwp \leftarrow Read()$
      In Parallel
        Process *1*
          if τ_l *and* $\tau_r = 0$ **then** // No zones in Right or Left
            $Mr_{rl} \leftarrow 0$
          else if $\tau_l = \tau_r$ **then** // Right/Left zones are equal
            if $y_w > y_d$ **then**
              $Mr_{rl} \leftarrow \tau_r$ // Go Right
            else
              $Mr_{rl} \leftarrow \tau_l$ // Go Left
          else
            $Mr_{rl} \leftarrow max\ (\tau_l, \tau_r)$ // Choose wider free-zones
          $\phi \leftarrow CalcControl(Mr_{rl}, \tau_l, \tau_r)$ // See Algorithm 8
        Process *2*
          if τ_u *and* $\tau_d = 0$ **then** // No zones in UP or Down
            $Mr_{ud} \leftarrow 0$
          else if $\tau_u = \tau_d$ **then** // Up/Down zones are equal
            if $z_w > z_d$ **then**
              $Mr_{ud} \leftarrow \tau_u$ // Go Up
            else
              $Mr_{ud} \leftarrow \tau_d$ // Go Down
          else
            $Mr_{ud} \leftarrow max\ (\tau_u, \tau_d)$
          $\vartheta \leftarrow CalcControl(Mr_{ud}, \tau_u, \tau_d)$ // See Algorithm 8
      $X \leftarrow \text{Nav}(\phi, 1, 0, \vartheta)$
    else
      $X \leftarrow \text{Nav}(0, 0, 0, 0)$
    goto Algorithm 3 // See Algorithm 3 (Chapter 4)

Algorithm 8: Calculate Avoidance Control

Input: Image zones $(\tau_l, \tau_r, \tau_u, \tau_d)$, Maneuver behavior (Mr_{rl}, Mr_{ud})
Output: Roll ϕ, Vertical speed ϑ
Define: Control gains ($K_{p\phi}$, $K_{i\phi}$, $K_{d\phi}$, $K_{p\vartheta}$, $K_{i\vartheta}$, $K_{d\vartheta}$), UAV position $X = (x_d, y_d, z_d)$, Loop interval time dt, Integral error $error_{integral}$ and Previous error pre_error
Calculate error:
    $error_u \leftarrow z_d + \left(\frac{h_{UAV}}{2}\right) + 20\ cm$ $error_d \leftarrow z_d - \left(\frac{h_{UAV}}{2}\right) + 20\ cm$
    $error_r \leftarrow y_d + \left(\frac{w_{UAV}}{2}\right) + 20\ cm$ $error_l \leftarrow y_d - \left(\frac{w_{UAV}}{2}\right) + 20\ cm$
In Parallel
    Process *1*
        if $Mr_{rl} = \tau_l$ **then**
            $error_{integral} + \leftarrow error_l \times dt$
            $\phi \leftarrow (K_{p\phi} \times error_l) + (K_{i\phi} \times error_{integral}) + (K_{d\phi} \times \frac{error_l - pre\_error}{dt})$
            $pre\_error \leftarrow error_l$
        else if $Mr_{rl} = \tau_r$ **then**
            $error_{integral} + \leftarrow error_r \times dt$
            $\phi \leftarrow (K_{p\phi} \times error_r) + (K_{i\phi} \times error_{integral}) + (K_{d\phi} \times \frac{error_r - pre\_error}{dt})$
            $pre\_error \leftarrow error_r$
        else
            $\phi \leftarrow 0$
        if $\phi > max$ **then**
            $\phi \leftarrow max$
        else if $\phi < min$ **then**
            $\phi \leftarrow min$
    Process *2*
        if $Mr_{ud} = \tau_u$ **then**
            $error_{integral} + \leftarrow error_u \times dt$
            $\vartheta \leftarrow (K_{p\vartheta} \times error_u) + (K_{i\vartheta} \times error_{integral}) + (K_{d\vartheta} \times \frac{error_u - pre\_error}{dt})$
            $pre\_error \leftarrow error_u$
        else if $Mr_{ud} = \tau_d$ **then**
            $error_{integral} + \leftarrow error_d \times dt$
            $\vartheta \leftarrow (K_{p\vartheta} \times error_d) + (K_{i\vartheta} \times error_{integral}) + (K_{d\vartheta} \times \frac{error_d - pre\_error}{dt})$
            $pre\_error \leftarrow error_d$
        else
            $\vartheta \leftarrow 0$
        if $\vartheta > max$ **then**
            $\vartheta \leftarrow max$
        else if $\vartheta < min$ **then**
            $\vartheta \leftarrow min$
$X \leftarrow$ EstimateUAVPose( ) `// See Algorithm 1 (Chapter 3)`

Finally, by estimating the new UAV after avoidance, the algorithm recalculates the new waypoints in order the UAV to be able to return back to its predefined path and activate the detection process.

In the case that the *AreaScale* is greater than **2** and the *SizeScale* of the keypoints is greater than **1.5**, a "*Hover*" command is sent to the UAV. That is because if the ratios exceed these limits, this means that the obstacle is very close to the UAV (less than 50 *cm*), as it mentioned in Chapter 4 and shown in Figure 4.8.

5.4 Results (*Obstacle Avoidance*)

One of the most important aspects in the avoidance phase is based on the robust control system for the UAV, at which it is necessary to know its dynamic model as it is explained in Appendix A. However, to avoid the complexity in modeling, the control was applied over the internal control of the system, modifying the *roll* and the *vertical speed* in order to perform the maneuvers in y and z directions.

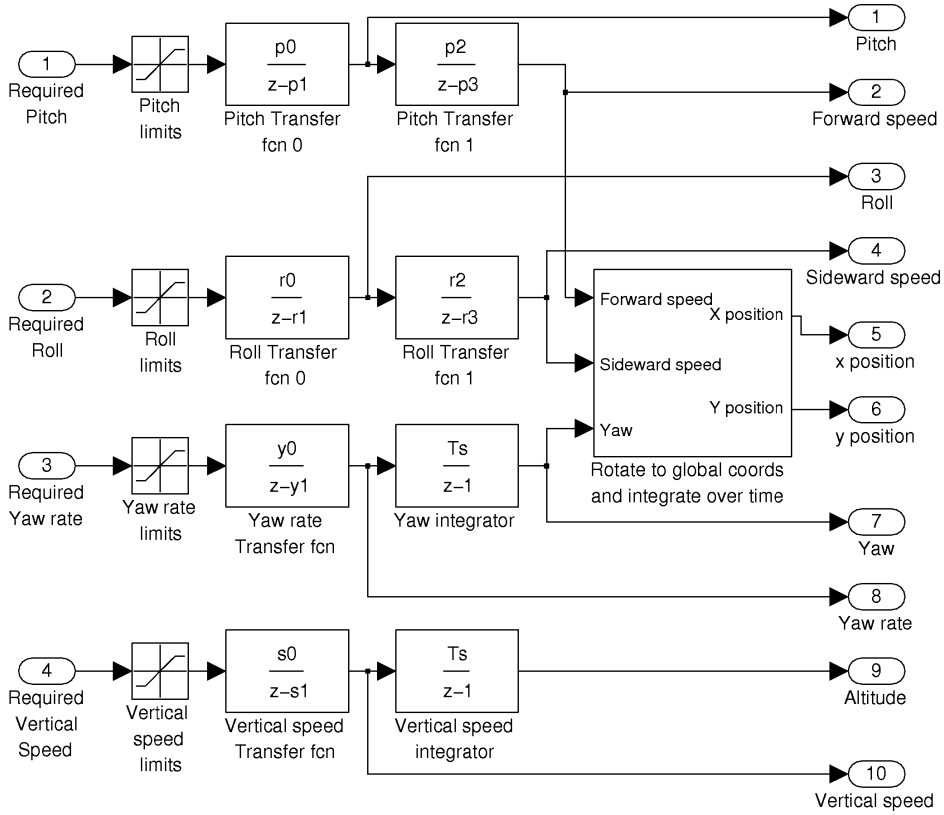
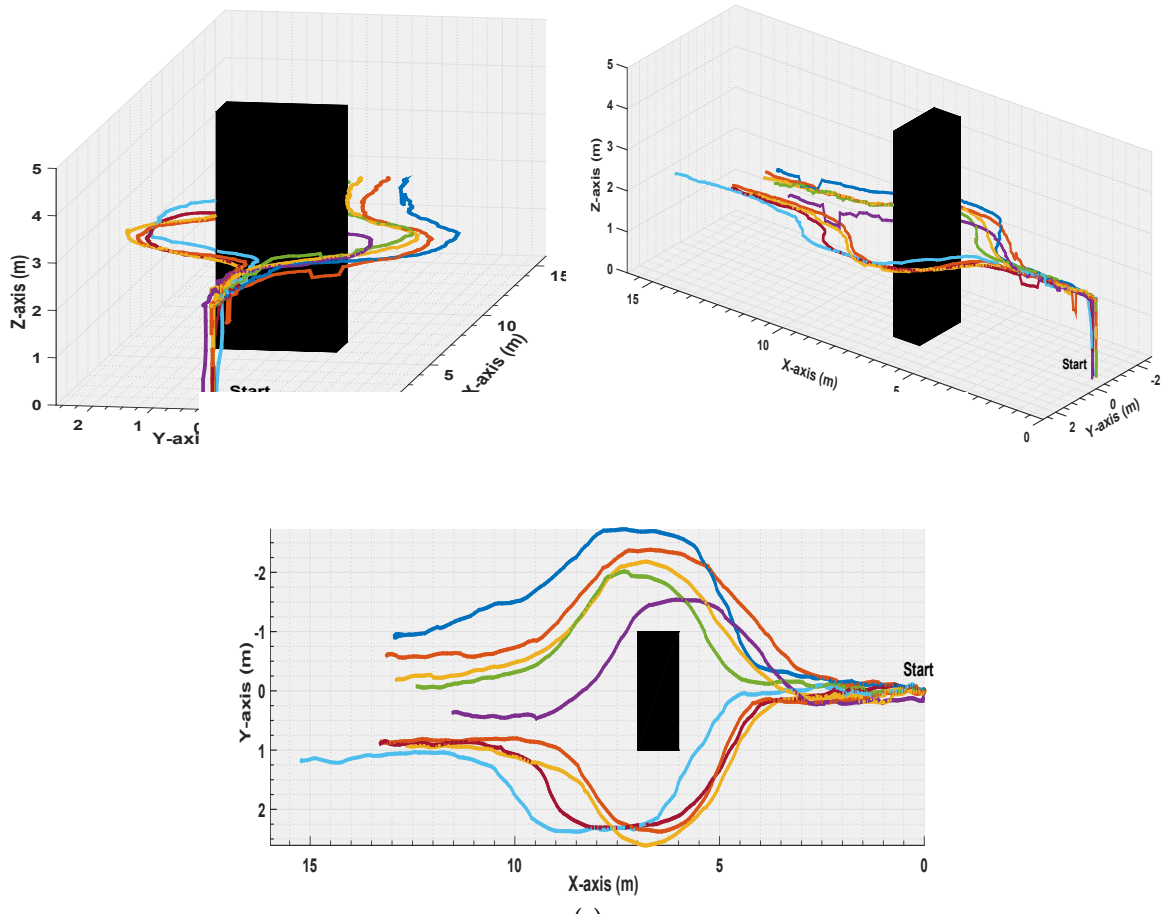


Figure 5.21 UAV Internal control structure [170]

Figure 5.21 shows the internal control of the UAV which maintains the stability of the flight. This control system is governed by the inputs (*roll*, *pitch*, *yaw* angles, and *vertical speed*), therefore the implemented controller realizes the UAV actual position, orientation and velocity.

In order to evaluate the obstacle avoidance algorithm, different obstacles are situated in different paths in indoor and outdoor experiments, of a total number of 1000 detected obstacles, and with different flight velocities.

These experiments are the same that are carried out to evaluate the obstacle detection algorithm, so that, the experiments are performed in two scenarios: The first one is the predefined straight flight, which aims to evaluate the avoidance algorithm in a motion. The second scenario is the hover stability flight. Where the obstacles approach to the UAV while it flies in the hover flight mode.

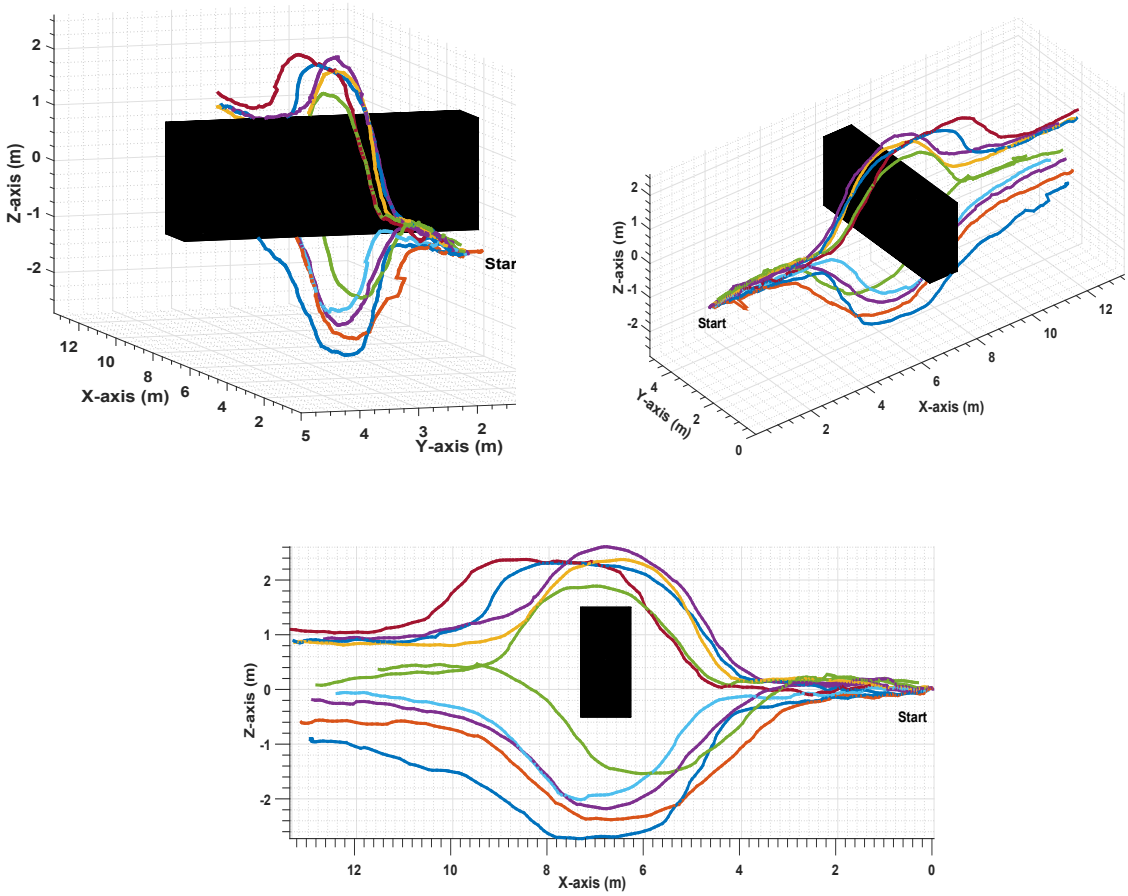


(c)

Figure 5.22 *Left-Right* Avoidance Maneuver, 9 experiments; **a:** Front view, **b:** 3D perspective-view, and **a:** 2D top-view.

Figures 5.22 and 5.23 demonstrate an example of a set of experiments presenting the first scenario. In these experiments, the UAV is flying in a velocity of $2m/s$. All the flights started from the same *start* point, and during the flight, an obstacle is situated in the UAV path. Figure 5.22 illustrates the UAV ability to perform avoidance maneuvers in the Left or Right directions of a total number of 9 experiments.

Similarly, in Figure 5.23, the success in avoiding hanged obstacles performing vertical maneuvers in the z direction, by passing above and under the obstacle in a total number of 10 experiments, is represented.



(c)

Figure 5.23 *Up-Down* Avoidance Maneuver, 10 experiments; **a:** Front view, **b:** 3D perspective-view, and **a:** 2D side-view.

From the results, it has been shown that the UAV takes the avoidance action in real-time once the obstacle is detected within a range of distances between 90 and 120 cm.

Finally, Table 5.9 shows a comparison of avoidance accuracy between two methods from the bibliography and the proposed algorithm. The accuracy results display that the best performance belongs to the proposed algorithm reaching 93% of accuracy.

Table 5.9 Comparison of Avoidance Accuracy.

Algorithm	**Total**	**Success**	**Failure**	**Accuracy (%)**
SURF + Template matching [228]	23	20	3	87.0
Relative Distance Estimation [261]	35	31	4	88.57
Proposed Algorithm	100	93	7	93.0

5.5 Summary

This chapter presented the combination mission of the waypoint tracking coupled with the collision avoidance control system; in order to achieve the autonomous navigation for the UAVs in complex environments.

For this purpose, on one hand, Fuzzy Logic Control (FLC) is formulated to control the stability of the UAV in the *Hover* flight mode, as well as to track and follow the predefined waypoints. Two parallel controllers are implemented; the first one is to control the heading angle and the the altitude of the UAV, whilst the second one is implemented to control the angles (*pitch*, *roll*), and the vertical velocities in (x, y)-axes.

The performance and the robustness of the presented algorithms are clearly stated from the results that are obtained from several experiments in both indoor and outdoor environments.

First, it is illustrated that the UAV has the ability to follow the predefined waypoints, with minimal error ($\pm$30 cm in *(x,y)*-direction and $\pm$5 cm in the z-direction), in booth indoor and outdoor arenas.

Second, the results show that the UAV is able to avoid any frontal collision within a rang of distances between 90 and 120 *cm*, with high level of accuracy of 93%.

The strengths of the presented applications are clearly stated in this dissertation, where the algorithms demonstrate hight accuracy in the tracking g the waypoints and avoiding the collisions. However, the specific drawbacks that should be taken into account are mainly related with the nature of the sensing devices used, that is, the monocular camera has the drawback of the high sensitivity to lighting conditions; such as direct sun light may lead to lack of information.

CHAPTER 6

Conclusions and Future Work

Visual sensors or cameras provide rich information about the surrounding environment. This advantage makes the cameras to be promising sensors, which can be used with the Unmanned Aerial Vehicles (UAVs) as the main sensor, or complementing other sensors; in order to improve the accuracy and the robustness of sensing the surrounding environment and perform the correct maneuvers to achieve its assigned tasks in the different applications.

In addition, computer vision algorithms and techniques; such as feature detection and tracking, image segmentation and triangulation are implemented to understand and process this information in order to provide several number solutions, that can be applied in the applications; such as navigation systems, map reconstruction (*2D* or *3D*), surveillance and inspections.

6.1 Conclusions

This dissertation presents a navigation system based on vision algorithms, which is applied on the autonomous UAVs. In the literature review in Chapter 2, it has been studied the advances and progression of the use the visual sensors and the computer vision algorithms in different applications -from basic to complex tasks that need high levels of accuracy and flight stabilization.

The main contribution of this dissertation is to present novel real-time solutions based on monocular cameras, that are mounted on-board on the UAV; in order to provide a robust navigation system for indoor and outdoor applications. The proposed solutions deal with the problems of **pose estimation**, **obstacle detection**, **collision avoidance decision**, and **control systems** in order to enhance the autonomy of the flights.

The proposed algorithms in this dissertation are summarized as follows:

- **3D pose estimation based on visual odometry**
  - The UAV 3D pose estimation algorithm is developed and validated. The proposed algorithm is based on the feature detection and tracking techniques using on-board monocular downward-looking camera.

    After performing an evaluation study on the most reliable detection and description techniques, the proposed algorithm used a combination of SIFT detector and FREAK descriptor. This SIFT-FREAK combination has the advantage of increasing the performance and the robustness as well as decreasing the computational time of the feature detection and description step.

    In addition, the algorithm calculates the *world-to-frame* and *frame-to-frame* homographies. Thereafter, by decomposing these homographies to its translation and rotation matrices, the 3D pose (x,y,z) and (ϕ,θ,ψ) of the UAV is estimated.

    Several experiments have been performed in order to validate the performance of the proposed algorithm in both indoor and outdoor scenarios, and the obtained results proved the accuracy of the algorithm is 99.4% with an overall performance of 50 *cm* and 30 *cm* of overall error for outdoor environment and 35 *cm* and 20 *cm* in indoor scenarios on the *(x,y)-axes* respectively.

- **Size-expansion monocular vision-based obstacle detection**

  - In this dissertation, a bio-inspired approach using monocular camera is presented in order to mimic the human behavior of obstacle detection, and to be applied on UAVs for real-time applications. In which, instead of reconstructing the 3D models of the objects, the presence of approaching obstacles is estimated based on the change in the size property of the detected features points (diameter), combined with the size (area) of the constructed convex hull from these detected points from the consecutive frames during the motion of the UAV.

    The proposed algorithm has been evaluated by performing different experiments in indoor and outdoor environments. The results show that the algorithm is able to detect the approaching obstacles in a range of distances between 90 and 120 *cm* with a total accuracy of 97.4%.

- **Path planning, waypoint tracking and obstacle avoidance**

  - To achieve the purposes of the autonomous flights, this dissertation presented control systems that are based on the visual information for path planning and waypoint tracking as well as obstacle avoidance. a generic optimization approach is applied to solve the path planning problem. The algorithm is optimized by SA metaheuristic technique to reach the minimum traveling path from starting position to the goal position. The output path is divided into several waypoints. Then the algorithm takes this vector of the generated waypoints and the current position of the UAV to send the control command to the UAV to follow and achieve the waypoints. At the same time, if an obstacle appeared in front of the UAV, an avoidance command is sent to perform the correct maneuver to avoid this obstacle. The main objective is the ability of the UAV to follow the waypoint generated from the path planning algorithm with minimal error and avoid any situated obstacle in this path.

From the previously proposed subsystem, this dissertation presented a navigation system which is based on visual information captured by on-board monocular cameras. This navigation system provides novel real-time and robust solutions using low-cost, lightweight and low power consumption visual sensors; in order to

enhance the stability and improve the autonomy of the UAV, for any complex task in indoor and outdoor missions.

6.2 Future Work

This dissertation has demonstrated that the visual sensors are considered as valuable sources and provide significant information, which can be used as the main source as well as a complementary with other systems, for the navigation purposes for the autonomous UAVs.

The proposed algorithms and approaches in this dissertation have been shown different solutions, with high levels of robustness and accuracy in real-time applications. However, in order to increase the performance of these algorithms and improving the robustness and the accuracy, different areas will be investigated in the future work.

Future researches, regarding intelligent aerial vehicles, are already in progress, and are focused on the following areas:

- Focusing on the improvement of the available perception systems into the aerial vehicle. This can be achieved by working with high-resolution sensors that can provide more rich information about the environments. In addition, optimizing the parallel processing techniques in order to allow the processing of these high-resolution sensors and minimizing the computational time.

- Future works will also focus on the fusion of the obtained visual information with the data provided by the other on-board sensors; such as GPS and IMU to enhance the estimation of the UAV pose according to the quality of the signal and the available information.

- Implementing algorithms that help to reconstruct 3D maps of the surrounding environments; keeping the use of the on-board monocular cameras.

- Increasing the environment understanding of the aerial vehicle, these new applications refer to obstacle understanding; such as pedestrian and building identification, among others, and the combinations of all this information with online context information; such as digital maps. The aerial environment understanding converges to the next generation of intelligent aerial vehicles,

which allows safe and reliable autonomous navigation of small aerial vehicles in future cities.

Generally, this dissertation focused on three main areas based on computer vision algorithms and approaches which are fundamental for any navigation systems for any autonomous UAV: Localization and Pose Estimation, Obstacle Detection and Avoidance Decision, and Visual Control; where each of these areas is a subject of more future research and improvements.

Publications

The work derived from this dissertation has been published in different international journals, international conferences, national conferences and patents.

International Journals

- **Waypoint Tracking, Guidance and Control for Autonomous Unmanned Aerial Vehicles (UAVs)**, Abdulla Al-Kaff, David Martín Gomez, Fernando García, Arturo de la Escalera, José María Armingol, Journal of Control Engineering Practice 2017. (Under revision)

- **Indoor and Outdoor Navigational System for UAVs based on Monocular Onboard Camera**, Abdulla Al-Kaff, Arturo de la Escalera, and José María Armingol, Journal of Intelligent & Robotic Systems, 2017. (Under revision).

- **Reactive Control System Design and Implementation for Unmanned Aerial Vehicles**, Abdulla Al-Kaff, Pasquale Longobardi, Fernando García. Journal of Advanced Robotics, 2017. (Under revision)

- **Survey of Computer Vision Algorithms and Applications for Unmanned Aerial Vehicles**, Abdulla Al-Kaff, David Martín, Fernando García, Arturo de la Escalera, and José María Armingol, Journal of Expert Systems With Applications, 2017. (In Press)

- **Obstacle Detection and Avoidance System Based on Monocular Camera and Size Expansion Algorithm for UAVs**, Abdulla Al-Kaff, Fernando García, David Martín, Arturo de La Escalera, and José María Armingol, Sensors, Volume 17, Issue 5, 2017.

- **IVVI 2.0: An intelligent vehicle based on computational perception**, David Martín, Fernando García, Basam Musleh, Daniel Olmeda, Gustavo Peláez, Pablo Marín, Aurelio Ponz, Cesar Rodríguez, Abdulla Al-Kaff, Arturo de la Escalera and José María Armingol, Expert Systems with Applications, Volume 41, Issue 17, 2014.

Book Chapters

- **ROS-based Approach for Unmanned Vehicles in Civil Applications**, Abdulla Al-Kaff, Francisco Miguel Moreno, and Ahmed Hussein, Robot Operating System (ROS) - The Complete Reference (Volume 3), 2017. (Under revision)

International Conferences

- **Understanding Rescue Environments using Multi-Object Tracking Algorithm**, Abdulla Al-Kaff, María José Gomez Silva, Francisco Miguel Moreno, Arturo de La Escalera, and José María Armingol, The International Conference on Robotics and Automation (ICRA18), May 21-25, 2018, Brisbane, Australia. (Under revision).

- **Intelligent Vehicle for Search, Rescue and Transportation Purposes**, Abdulla Al-Kaff, Francisco Miguel Moreno, Arturo de La Escalera, and José María Armingol, The 15th IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR 2017), October 11-13, 2017, Shanghai, China.

- **Aerial Image Mosaicking for Multi-purpose Civil Applications**, Abdulla Al-Kaff, Juan Camilo Soto Triviño, Raúl Sosa San Frutos, Arturo de la Escalera, and José María Armingol, Third Iberian Robotics Conference, November 22-24, 2017, Sevilla, Spain.

- **Homography-Based Navigation System for Unmanned Aerial Vehicles**, Abdulla Al-Kaff, Arturo de La Escalera, and José María Armingol, Advanced Concepts for Intelligent Vision Systems, September 18-21, 2017, Antwerp, Belgium.

- **VBII-UAV: Vision-based Infrastructure Inspection-UAV**, Abdulla Al-Kaff, Francisco Miguel Moreno, Luis Javier San José, Fernando García, David Martín, Arturo de la Escalera, Alberto Nieva, and José Luis Mena García, World Conference on Information Systems and Technologies, Springer, April 11-13, 2017, Porto Santo, Madeira, Portugal.

- **Monocular Vision-based Obstacle Detection/Avoidance for Unmanned Aerial Vehicles**, Abdulla Al-Kaff, Qinggang Meng, David Martín, Arturo de la Escalera, and José María Armingol, 2016 IEEE Intelligent Vehicles Symposium (IV). IEEE, June 19-22, 2016, Gothenburg, Sweden.

- **Autonomous Indoor Navigation of Low-cost Quadcopters**, Ahmed Hussein, Abdulla Al-Kaff, Arturo de la Escalera, and José María Armingol. 2015 IEEE International Conference on Service Operations And Logistics, And Informatics (SOLI), IEEE, November 15-17, 2015, Tunisia.

- **SIFT and SURF Performance Evaluation and the Effect of FREAK Descriptor in the Context of Visual Odometry for Unmanned Aerial Vehicles**, Abdulla Al-Kaff, Arturo de la Escalera, and José María Armingol. International Conference on Computer Aided Systems Theory, EUROCAST'15, Springer, February 10-15, 2015, Canary Islands, Spain.

National Conferences

- **Real-Time Image Mosaicking for Mapping and Exploration Purposes**, Abdulla Al-Kaff, Juan Camilo Soto Triviño, Raúl Sosa San Frutos, Arturo de la Escalera, and José María Armingol, XXXVIII Jornadas de Automática, September 6-8, 2017, Gijón, Spain.

Patents

- **A system and a method for surface aerial inspection**. European patent, Patent No. 16382353.7-1504.

Awards

- Best Computer Vision paper award in XXXVIII Jornadas de Automática, 2017.
- Co-author of Best Student Paper Award in the 10th International Conference on Service Operations and Logistics, and Informatics (SOLI2015, IEEE), 2015.

APPENDIX A

Quadcopter Model

A quadcopter is a rotary-wing UAV which is lifted by four rotors. In contrast to helicopters, quadcopters have a rigid compact structure without tail, in addition, it shares the same dynamics with the other rotary-wing UAVs (Hexacopter, Octocopter, etc.). However, it has low power consumption.

As it is shown in Figure A.1, the quadcopter has two coordinate frames; First, a right hand inertial frame (***Earth frame***) denoted by $\boldsymbol{A}$, with unit vectors along its axes defined as $\begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}$. At which, the position of the center of mass of the quadcopter is defined relative to the inertial frame. Second, a right hand body frame is used to define the quadcopter orientation. The body frame is denoted by $\boldsymbol{B}$ with unit vectors along of its axes by $\begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}$.

The quadcopter has different two configurations; **Cross (×)** and **Plus (+)** configuration, where the orientation of the body frame axes varies between both configurations. As it is shown in Figure A.2, in plus configuration *x-axis* lies along arm of motor one, and *y-axis* lies along arm of motor two, and *z-axis* is directed upward.

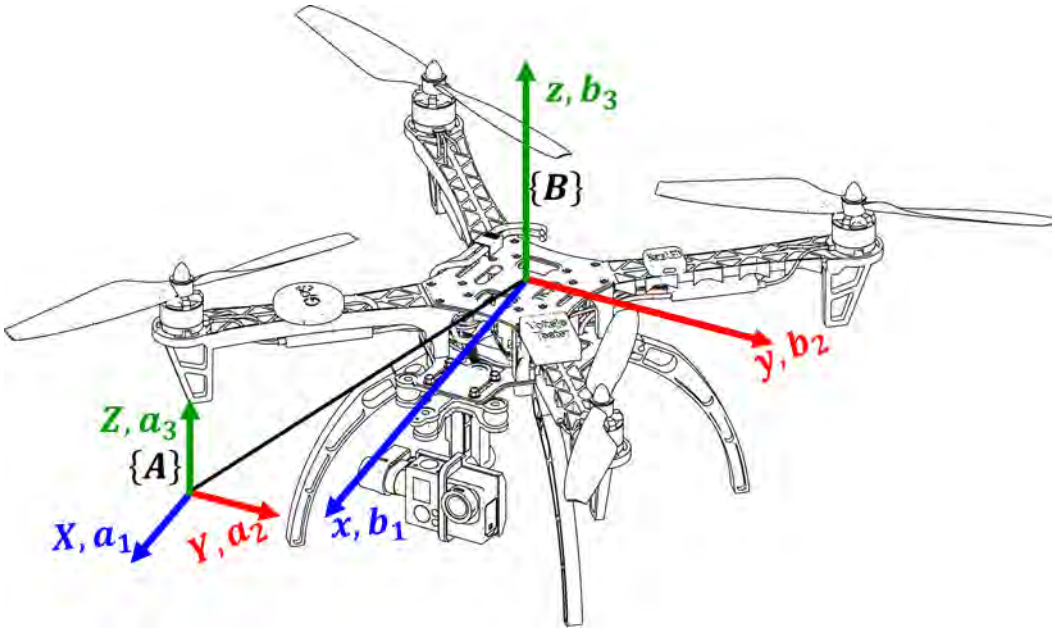


Figure A.1 different quadcopter frames

Both motor one and two spin in opposite direction and are placed at an equivalent distance d from the center of mass of the quadcopter.

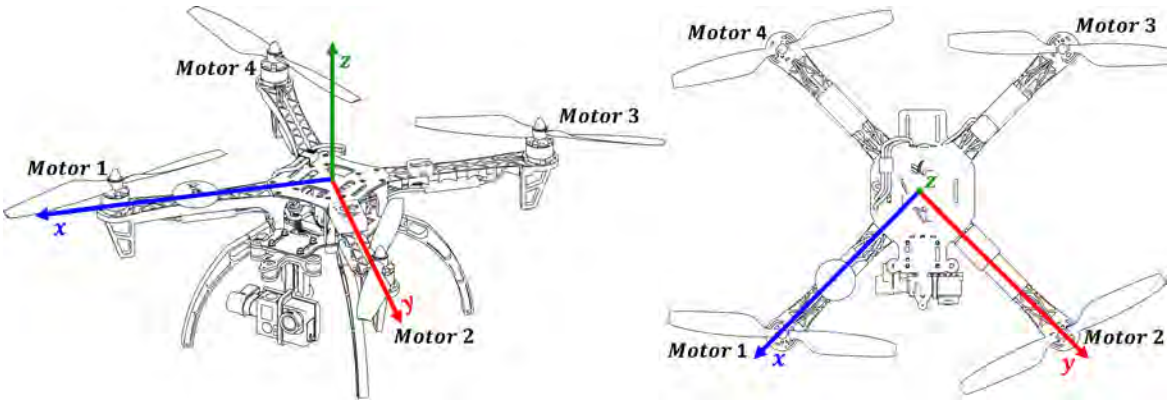


Figure A.2 plus quadcopter

In cross configuration, x and y planes are rotated with an angle of 45 degrees in the positive *yaw* direction. *x-axis* lies symmetrically between motor one and two, *y-axis* lies symmetrically between motor two and three and *z-axis* is directed upward, as it is shown in Figure A.3.

The origin of the global frame is defined as the starting point on the ground where the quadcopter takes off. The position of the quadcopter is defined by the position vector between the origin of the body frame and the origin of the global

frame.

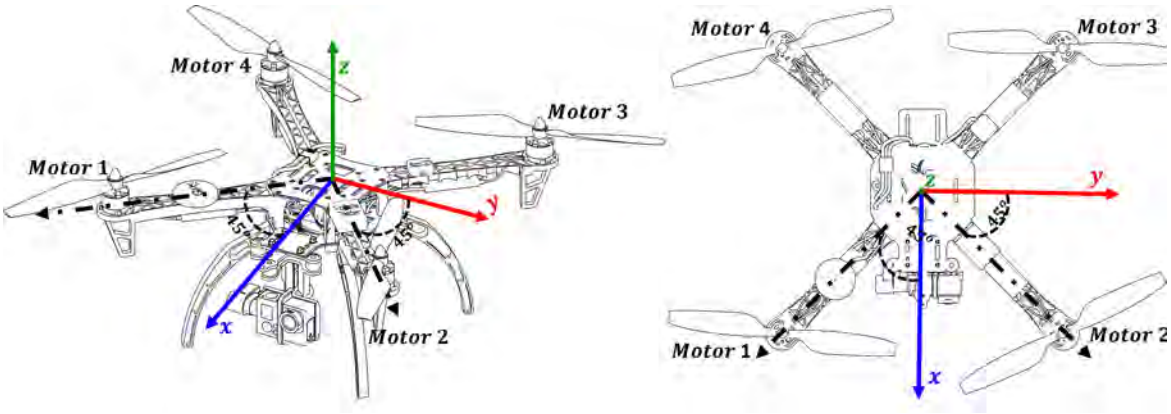


Figure A.3 x quadcopter

The quadcopter has six degrees of freedom: three rotary motions (*roll*, *pitch* and *yaw*) (Figure A.4), and three translation motions in x, y and z axes. *Roll* (ϕ) is the rotation on the *x-axis* of the body frame, *pitch* (θ) is the rotation on the *y-axis* of the body frame, and *yaw* (ψ) is the rotation on the body frame *z-axis*.

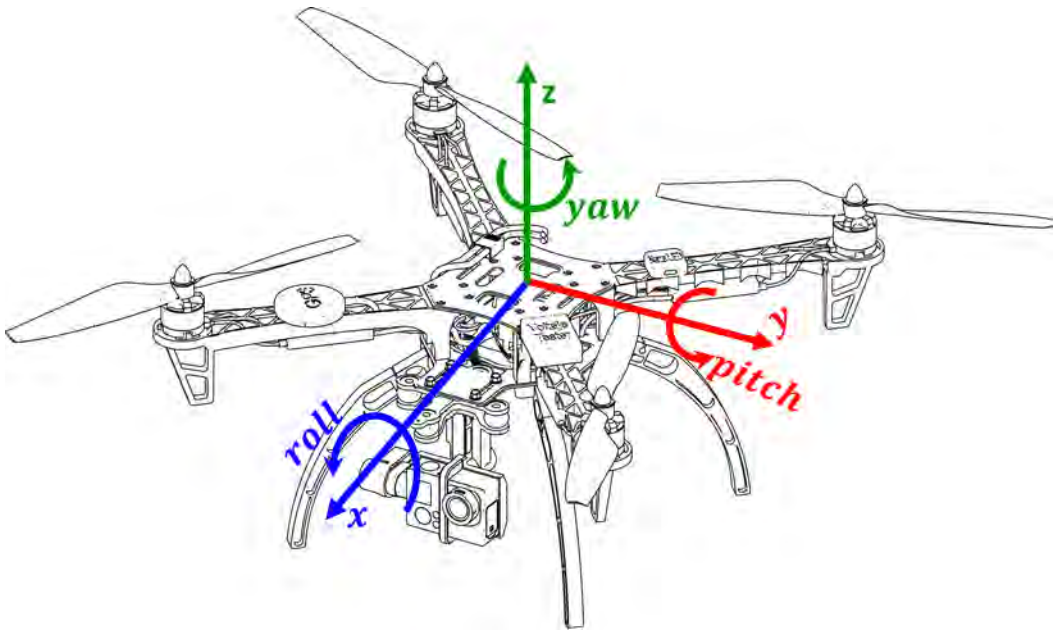


Figure A.4 quadcopter angles

A.1 Mathematical Model

Before starting with the mathematical model, the used notation method should be discussed. The notation method is developed to give sufficient description of critical variables in the complex six degrees of freedom systems. An example of the notations is shown below:

$$^{b}\dot{\mathrm{V}}^{b}_{CM|i} \tag{A.1}$$

The base variable $\dot{V}$ is the linear acceleration. The top left superscript b states that the derivative is done relative to the body frame, the right top superscript b states that the acceleration is represented by the body frame vector components. The subscript $CM|i$ states that the acceleration is relative to the center of mass with respect to the inertial frame.

A.1.1 Moment of Inertia

The first element of the system model defines the inertia matrix. The inertia matrix is an important parameter to the quadcopter flight dynamics, it defines the mass moment of inertia of the system across each axis. The inertia matrix is calculated by:

- Breaking the quadcopter into separate components (central hub, arms, ESCs, rotors and battery), and modeling each component as a simplified geometric shape of constant density.
- Measure the weight of each component, and estimate the total weight.
- Determine the contribution of each component in moment of inertia of the x, y and z axes using parallel axis theorem.
- The summation of the contribution of each component produces the total moment of inertia matrix. The obtained moment of inertia matrix has a diagonal form for either plus or cross configuration, due to the quadcopter symmetry.

$$J^{b} = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \tag{A.2}$$

where, J^b is the body frame inertia matrix of the quadcopter, with J_{xx}, J_{yy} and J_{zz} being the inertia across each axis. The diagonal form of the equation simplifies the inversion calculations to be used in angular velocity state equation.

A.1.2 Motor Dynamics

The motors thrust is the driving force of the quadcopter in order to provide its lift. The thrust generated by the rotor provides a normal force to x and y planes, directed in positive **Z** direction. The thrust $\boldsymbol{T}$ of a single motor in the system can be calculated as follows:

$$T = C_T \rho A_R r^2 \varpi^2 \tag{A.3}$$

where C_T is the thrust coefficient for a specific rotor, ρ is air density, A_R is the area of propeller rotation, r^2 and ϖ^2 are the radius of the rotor and its angular velocity respectively. To simplify the flight modeling a lumped parameter approach is used:

$$T = c_T \varpi^2 \tag{A.4}$$

where c_T is the lumped parameter thrust coefficient of each motor.

The torque force of the motor is studied to understand the *yaw* motion of the quadcopter. The torque can be calculated in a similar manner to the thrust based on lumped parameter approach. The torque equation is shown as follows:

$$Q = c_Q \varpi^2 \tag{A.5}$$

where Q is the torque provided by single motor, and c_Q is the lumped torque coefficient.

The thrust and the torque are directly proportional to the Revolutions per minute (RPM) of the motor, and can not be directly controlled by the control system output (throttle command). Therefore, linear regression is used to convert the throttle command to RPM of the motors. The linear regression equation is shown below:

$$\varpi_{ss} = (Throttle\%)\, c_R + b \tag{A.6}$$

where ϖ_{ss} is the steady-state motor RPM, $Throttle\%$ is the throttle command percentage, c_R is the conversion coefficient from $Throttle\%$ to RPM, and b is the y-intercept of the linear regression equation.

A.1.3 Initial Matrix Construction

The total thrust of the quadcopter is defined as the sum of the individual thrust of each motor as follows:

$$\sum T = \sum_{i=1}^{N} |T_i| = c_T \left(\sum_{i=1}^{N} \omega_i^2 \right) \tag{A.7}$$

The moment applied by the quadcopter, and obtained from motor forces and air resistance can be calculated by the following equations:

$$\tau_\phi = c_T \sum_{i=1}^{N} d_i sin(\alpha_i) \omega_i^2 \tag{A.8}$$

$$\tau_\theta = -c_T \sum_{i=1}^{N} d_i cos(\alpha_i) \omega_i^2 \tag{A.9}$$

$$\tau_\psi = c_Q \sum_{i=1}^{N} \sigma_i \omega_i^2 \tag{A.10}$$

where α_i is the angle between the motor arm and the positive body frame axis, and ω_i is the direction of motor rotation.

Based on the derived thrust and torque equations, an initial matrix describing the quadcopter dynamics can be constructed. In the *plus* configuration, the initial matrix is represented as follows:

$$\begin{bmatrix} \sum T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ 0 & d_+ c_T & 0 & -d_+ c_T \\ -d_+ c_T & 0 & d_+ c_T & 0 \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{A.11}$$

where d_+ is the length of the arm from quadcopter hub center to motor.

In *cross* configuration, $d_\times$ is used instead, and it is defined as the distance between the central hub and the system axes of rotation, where $d_\times = d_+ sin(45)$. Since the *z-axis* is the same between both configuration, the *yaw* torque acting in the

z-axis direction is the same, where:

$$\begin{bmatrix} \sum T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ -d_x c_T & d_x c_T & d_x c_T & -d_x c_T \\ -d_x c_T & -d_x c_T & d_x c_T & d_x c_T \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix} \begin{bmatrix} \varpi_1^2 \\ \varpi_2^2 \\ \varpi_3^2 \\ \varpi_4^2 \end{bmatrix} \tag{A.12}$$

A.1.4 Gyroscopic Forces

Gyroscopic precession is an extra force to be taken into account before creating moment matrix. Gyroscopic precession takes place due to the change in the axis of rotation of a rotating body, in addition, it depends on the inertia of the motors rotating components (J_m), the angular rates of *roll* and *pitch* (P and Q), and the angular speed of each motor (ϖ_i). The gyroscopic torques are created for *roll* and *pitch* and have the following equation:

$$\tau_{\phi_{gyro}} = J_m Q \left(\frac{\pi}{30}\right) (\varpi_1 - \varpi_2 + \varpi_3 - \varpi_4) \tag{A.13}$$

$$\tau_{\theta_{gyro}} = J_m P \left(\frac{\pi}{30}\right) (-\varpi_1 + \varpi_2 - \varpi_3 + \varpi_4) \tag{A.14}$$

where the $\frac{\pi}{30}$ term appears due to the conversion from RPM to radians.

A.1.5 Final Matrix Construction

Quadcopter is balanced under the effect of the gravity and the lift of the rotors. The obtained force is oriented in the positive Z direction and can be represented as follows:

$$F_{A,T}^b = \begin{bmatrix} 0 \\ 0 \\ C_T \left(\varpi_1^2 + \varpi_2^2 + \varpi_3^2 + \varpi_4^2\right) \end{bmatrix} \tag{A.15}$$

where $F_{A,T}^b$ refers to the forces applied to the body frame on the quadcopter due to aerodynamics and thrust. The final moment matrix takes into account the previously mentioned aerodynamic, gyroscopic and thrust moments of the quadcopter. Based on the configuration, the final matrix can be constructed; in the case of the

plus configuration as follows:

$$M^b_{A,T} = \begin{bmatrix} d_+c_T\omega_2^2 - d_+c_T\omega_4^2 + J_mQ\left(\frac{\pi}{30}\right)(\omega_1 - \omega_2 + \omega_3 - \omega_4) \\ -d_+c_T\omega_1^2 + d_+c_T\omega_3^2 + J_PQ\left(\frac{\pi}{30}\right)(-\omega_1 + \omega_2 - \omega_3 + \omega_4) \\ -c_Q\omega_1^2 + c_Q\omega_2^2 - c_Q\omega_3^2 + c_Q\omega_4^2 \end{bmatrix} \tag{A.16}$$

In the case of the *cross* configuration, the final matrix is represented by:

$$M^b_{A,T} = \begin{bmatrix} -d_xc_T\omega_1^2 + d_xc_T\omega_2^2 + d_xc_T\omega_3^2 - d_xc_T\omega_4^2 + J_mQ\left(\frac{\pi}{30}\right)(\omega_1 - \omega_2 + \omega_3 - \omega_4) \\ -d_xc_T\omega_1^2 - d_xc_T\omega_2^2 + d_xc_T\omega_3^2 + d_xc_T\omega_4^2 + J_PQ\left(\frac{\pi}{30}\right)(-\omega_1 + \omega_2 - \omega_3 + \omega_4) \\ -c_Q\omega_1^2 + c_Q\omega_2^2 - c_Q\omega_3^2 + c_Q\omega_4^2 \end{bmatrix} \tag{A.17}$$

This model assumes static thrust and torque, where $M^b_{A,T}$ is the moment of the body frame. Furthermore, it neglects some additional effects; such as drag force on the frame and blade flapping.

A.1.6 State Equations

The angular velocity state equation describes the change in *roll* (P), *pitch*(Q) and *yaw* (R) rates of the quadcopter. This equation takes into consideration the inertia, moments applied, and angular velocity. The angular velocity state equation is represented as follows:

$${}^b\dot{\omega}^b_{b|i} = \left(J^b\right)^{-1}\left[M^b_{A,T} - \Omega^b_{b|i}J^b w^b_{b|i}\right] = \begin{bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix} \tag{A.18}$$

where ${}^b\dot{\omega}^b_{b|i}$ is the angular acceleration in the body frame on each axis with respect to inertial frame, and is defined as:

$${}^b\dot{\omega}^b_{b|i} = \begin{bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix} \tag{A.19}$$

where $\Omega^b_{b|i}$ is the cross product matrix for rotational velocity which can be written as:

$$\Omega^b_{b|i} = \begin{bmatrix} 0 & -R & Q \\ R & 0 & -P \\ -Q & P & 0 \end{bmatrix} \tag{A.20}$$

where $w^b_{b|i}$ is the rotational velocity of the system in the body frame and is directly defined by P, Q and R:

$$w^b_{b|i} = \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \tag{A.21}$$

Using Euler Kinematic Equation, the rate of change of the Euler angles within the inertial frame is estimated as follows:

$$\dot{\varepsilon} = H(\varepsilon)\, w^b_{b|i} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{A.22}$$

where, ε is the Euler angles matrix where, $\varepsilon = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}$

Before defining the function $H(\varepsilon)$, the rotation matrix should be discussed. Rotation matrix is a matrix that transfers the motion of the quadcopter from the body frame to a new reference frame. The rotation matrix is used here in order to transfer from body frame to inertial frame, and can be found by matrix multiplication of the three rotation matrices corresponding to each angle.

$$u^b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & s(\phi) \\ 0 & -s(\phi) & c(\phi) \end{bmatrix} \begin{bmatrix} c(\theta) & 0 & -s(\theta) \\ 0 & 1 & 0 \\ s(\theta) & 0 & c(\theta) \end{bmatrix} \begin{bmatrix} c(\psi) & s(\psi) & 0 \\ -s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} u^i \tag{A.23}$$

where $c(\phi)$ and $s(\theta)$ represent the sine and cosine of the angles respectively.

Using aerospace rotation sequence, the rotation matrix that transfers from the

inertial to the body frame can be estimated, and has the following form:

$$C_{b|i} = \begin{bmatrix} c(\theta)c(\psi) & c(\phi)s(\psi) & -s(\theta) \\ -c(\phi)s(\psi) + s(\phi)s(\theta)c(\psi) & c(\phi)c(\psi) + s(\phi)s(\theta)s(\psi) & s(\phi)c(\theta) \\ s(\phi)s(\psi) + c(\phi)s(\theta)c(\psi) & -s(\phi)c(\psi) + c(\phi)s(\theta)s(\psi) & c(\phi)c(\theta) \end{bmatrix} \quad \text{(A.24)}$$

Therefore, the angular velocity of the quadcopter in the body frame are related to the change in angle rotation as shown as follows:

$$w^b_{b|i} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + c_\phi \left(\begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + c_\theta \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \right) \quad \text{(A.25)}$$

Using matrix multiplications, additions and taking the derivatives of the kinematic equation, the rate of change of the Euler angles within the inertial frame equation can be derived as follows:

$$H(\varepsilon) = \begin{bmatrix} 1 & t(\theta)s(\phi) & t(\theta)c(\phi) \\ 0 & c(\phi) & -s(\phi) \\ 0 & \frac{s(\phi)}{c(\theta)} & \frac{c(\phi)}{c(\theta)} \end{bmatrix} \quad \text{(A.26)}$$

$$\dot{\varepsilon} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & t(\theta)s(\phi) & t(\theta)c(\phi) \\ 0 & c(\phi) & -s(\phi) \\ 0 & \frac{s(\phi)}{c(\theta)} & \frac{c(\phi)}{c(\theta)} \end{bmatrix} = H(\varepsilon) w^b_{b|i} \quad \text{(A.27)}$$

While this model produces effective results and creates a singularity when θ approaches to ± 90 degrees. Therefore, the accuracy and the stability of the model can be compromised at this angle ± 90 degrees. Considering our control intentions, this drawback will be avoided.

Another state equation is the velocity state equation. Based on the forces and accelerations acting on the quadcopter, this equation describes the acceleration of the center of mass of the system model as follows:

$${}^b\dot{\mathrm{V}}^b_{CM|i} = \left(\frac{1}{m}\right) F^b_{A,T} + g^b - \Omega^b_{b|i} w^b_{CM|i} = \begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{\omega} \end{bmatrix} \quad \text{(A.28)}$$

where ${}^{b}\dot{V}^{b}_{CM|i}$ is the linear acceleration in the body frame, acting at the center of mass of the quadcopter with reference to the inertial frame, m is the total measured mass of the quadcopter, and g^{b} the acceleration due to gravity acting in the body frame by the rotation matrix $c_{b|i}$, where:

$$g^{b} = c_{b|i} g^{i} \tag{A.29}$$

Based on this equation, the linear acceleration of quadcopter along X, Y and Z directions of the body frame can be determined. The last state equation to be discussed is the position state equation. The position state equation describes the linear velocity in the inertial frame of the center of mass of quadcopter.

$${}^{i}\dot{P}^{i}_{CM|i} = C_{i|b} V^{b}_{CM|i} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} \tag{A.30}$$

where $V^{b}_{CM|i}$ is the linear velocity of the quadcopter in the body frame, and it is transferred to the inertial frame using $C_{i|b}$.

APPENDIX B

Camera Model and Calibration

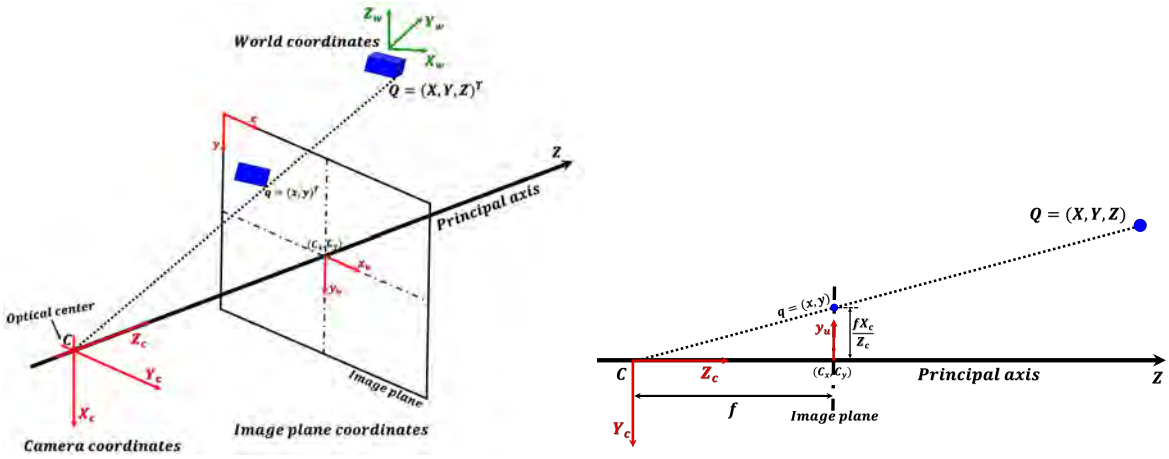
B.1 Camera Model

The purpose of defining the camera model is to map the 3-dimensional world coordinates system to a 2-dimensional coordinates system (image plane). The simplest camera model is the pinhole model (Figure B.1), at which, a point in the 3D space can be represented by $Q_w = (x_w, y_w, z_w)^T$, the point Q_w is projected to a 2D point $q_p = (x_p, y_p)^T$ onto the image plane by intersecting the ray – that connecting the 3D point Q_w and the camera center C - with the image plane, which is situated with a distance f (focal length in mm) from C, and $q_c = (x_c, y_c, z_c)^T$ is the point in the camera coordinates system.

B.1.1 Camera Intrinsic and Extrinsic Parameters

As it is illustrated in Figure B.1b, by looking at similar triangles, the point Q_w can be transformed from the world coordinate system into the point q_p in the image plane coordinated system as follows:

$$\frac{x_p}{f} = \frac{x_w}{z_w} \quad or, \quad \frac{y_p}{f} = \frac{y_w}{z_w} \tag{B.1}$$



(a) (b)

Figure B.1 Pinhole Camera Model; **a:** 3D and **b:** 2D.

By using the homogeneous coordinates, q_p can be expressed in a matrix notation, so that equation B.1 becomes as follows:

$$q_p = Kq_c$$
$$\lambda \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \tag{B.2}$$

where λ is the homogeneous scale factor and K is the calibration matrix that contains the intrinsic parameters; c_x and c_y are the coordinates of the center of the image in pixels, and f_x and f_y are the focal length in pixel units in X and Y directions

which calculated as follows:

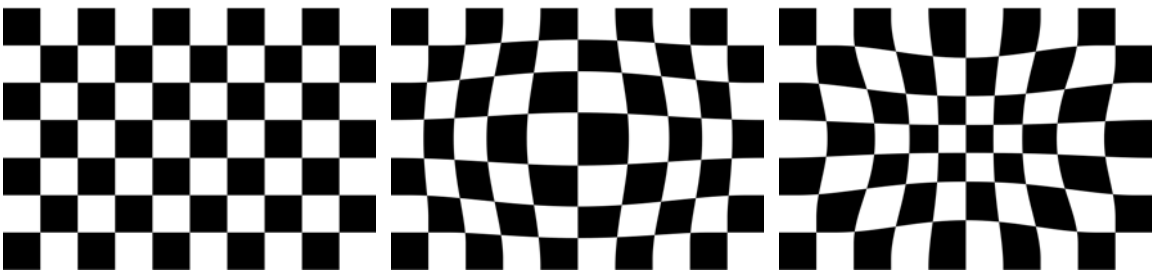
$$\begin{aligned} f_x &= f m_x \\ f_y &= f m_y \end{aligned} \tag{B.3}$$

where m_x and m_y are the number of pixels per unit distance.

In theory, camera lenses should not be affected by the geometric distortions. However, practically, the lenses are imperfect and suffer from a non-linear distortions effects. The "*Radial Distortion*", in which, the straight lines are mapped as curves, where the positions of the projected points in the image move in a radial direction from its real position in the world. The "*Tangential Distortion*" results that the parallel lines in real-world are not exactly parallel in the image plane. As shown in Figure B.2, two types of radial distortion can affect the images, the *Barrel* (positive radial distortion) (Figure B.2a) and the *Pincushion* (negative radial distortion) (Figure B.2b). The image distortion (radial and tangential) can be modeled as follows:

$$\begin{aligned} x_d &= x_u(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + (2p_1 x_u y_u + p_2(r^2 + 2x_u^2)) \\ y_d &= y_u(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + (2p_2 x_u y_u + p_1(r^2 + 2x_u^2)) \end{aligned} \tag{B.4}$$

where (x_d, y_d) and (x_u, y_u) are the distorted and undistorted coordinates respectively, (k_1, k_2, k_3) are the radial distortion coefficients, (p_1, p_2) are the tangential distortion coefficients, and $r = \sqrt{(x_u - x_c) + (y_u - y_c)}$ where (x_c, y_c) is the center of the radial distortion.



(a) (b) (c)

Figure B.2 Image Radial Distortions; **a:** No distortion, **b:** Barrel (positive radial distortion), and **c:** Pincushion (negative radial distortion).

On the other hand, the pose of the camera with respect to the world coordinates system can be expressed in terms of position and orientation, based on the camera extrinsic parameters (Figure B.3). In mathematical notation, the camera position is expressed by a vector T and the orientation by a matrix R.

Based on the pinhole model, the extrinsic parameters can be defined as follows:

$$
\begin{aligned}
q_c &= \left[R \mid T\right] Q_w \\
\lambda \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} &= \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}
\end{aligned}
\tag{B.5}
$$

From the equations B.2 and B.5, the camera parameters (intrinsic and extrinsic) can be represent as follows:

$$
q_p = K \left[R \mid T\right] Q_w \tag{B.6}
$$

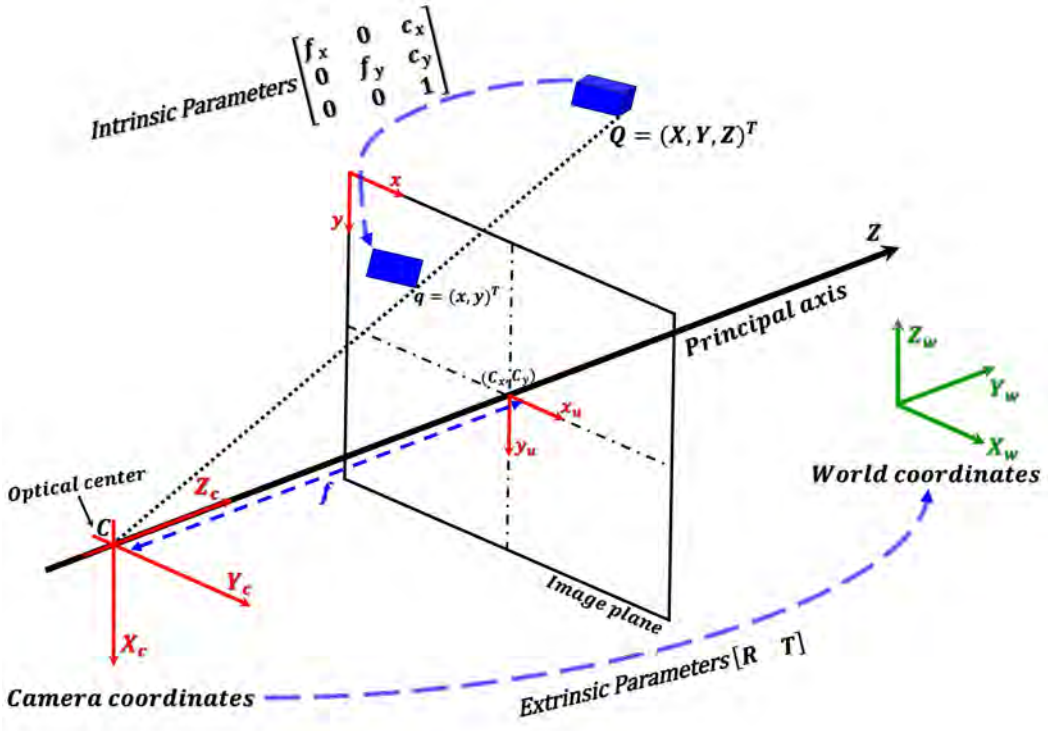


Figure B.3 Intrinsic and Extrinsic parameters

B.2 Camera Calibration

The calibration of any camera aims to estimate the intrinsic and extrinsic parameters that are required in order to calculate the metric information from the captured images [128, 349]. Intrinsic parameters contain the focal length in pixel units (f_x, f_y), the center of the image (c_x, c_y), the skew coefficient γ, the radial distortion coefficients (k_1, k_2, k_3) and the tangential distortion coefficients (p_1, p_2). Where the extrinsic parameters are the rotation matrix R and the translation vector T.

The estimation of the camera parameters is achieved by using the data of a planner pattern (chessboard), situated in different views and angles as shown in Figure B.4. First, the calibration step requires extracting the grid corners of the chessboard pattern in the images (the dimensions of the chessboard squares should be known in metric units). The corners extraction is preferably done manually one by one in order to calculate accurately the distortion coefficients.

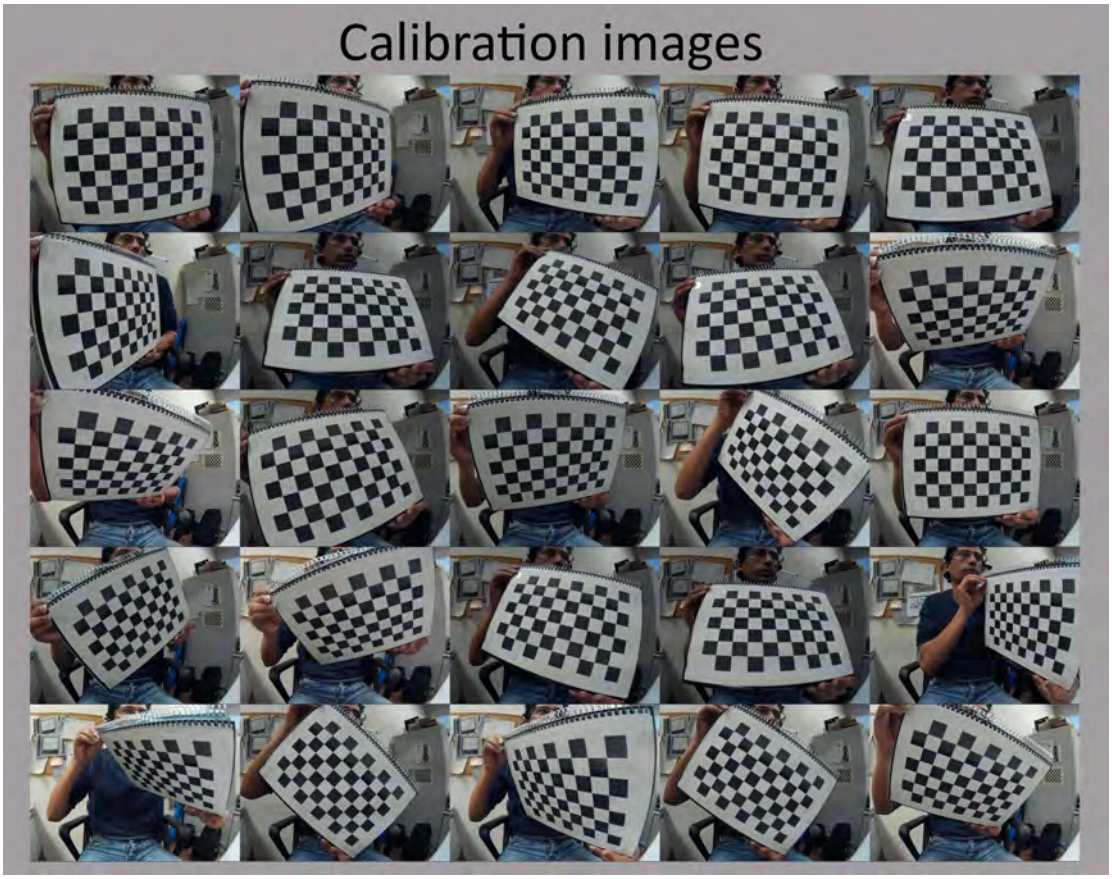


Figure B.4 Calibration images in different views.

By assuming that the planner pattern is on a distance $Z = 0$ in the world coordinates system, then equation B.6 is expressed as follows:

$$\lambda \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = K \begin{bmatrix} r_1 & r_2 & r_3 & T \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 0 \\ 1 \end{bmatrix}$$
$$\lambda \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = K \begin{bmatrix} r_1 & r_2 & T \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} \tag{B.7}$$

where $H = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix}$ is known as the homography matrix, which represents the relation of the point in the planner pattern $Q_w = (x_w, y_w, 1)^T$, and its image $q_p = (x_p, y_p, 1)^T$ where,

$$\lambda \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = H \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} \quad and \quad H = sK \begin{bmatrix} r_1 & r_2 & T \end{bmatrix} \tag{B.8}$$

where $s = \frac{1}{\lambda}$ is an arbitrary scalar. Because r_1 and r_2 are orthogonal, then:

$$h_1^T K^{-T} K^{-1} h_2 = 0 \tag{B.9}$$

$$h_1^T K^{-T} K^{-1} h_1 = h_2^T K^{-T} K^{-1} h_2 \tag{B.10}$$

To solve the camera calibration problem, an analytical process is performed, in which, an initial estimation followed by a nonlinear optimization based on the maximum likelihood criterion are implemented. Suppose,

$$A = K^{-T} K^{-1} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{12} & A_{22} & A_{23} \\ A_{13} & A_{23} & A_{33} \end{bmatrix} \tag{B.11}$$

where A is a symmetric matrix of elements that contain the intrinsic parameters:

$$A = \begin{bmatrix} \frac{1}{f_x^2} & 0 & \frac{-c_x}{f_x^2} \\ 0 & \frac{1}{f_y^2} & \frac{-c_y}{f_y^2} \\ \frac{-c_x}{f_x^2} & \frac{-c_y}{f_y^2} & \frac{-c_x}{f_x^2} + \frac{-c_y}{f_y^2} + 1 \end{bmatrix} \tag{B.12}$$

and can be defined as 6D vector as follows:

$$a = \begin{bmatrix} A_{11} & A_{12} & A_{22} & A_{13} & A_{23} & A_{33} \end{bmatrix}^T \tag{B.13}$$

Supposing that the i^{th} column vector of H is $h_i = \begin{bmatrix} h_{i1} & h_{i2} & h_{i3} \end{bmatrix}^T$, then,

$$h_i^T B h_j = \begin{bmatrix} h_{i1}h_{j1} \\ h_{i1}h_{j2} + h_{i2}h_{j1} \\ h_{i2}h_{j2} \\ h_{i3}h_{j1} + h_{i1}h_{j3} \\ h_{i3}h_{j2} + h_{i2}h_{j3} \\ h_{i3}h_{j3} \end{bmatrix}^T \begin{bmatrix} A_{11} \\ A_{12} \\ A_{22} \\ A_{13} \\ A_{23} \\ A_{33} \end{bmatrix}^T = v_{ij}^T b \tag{B.14}$$

where, $v_{ij} = \begin{bmatrix} h_{i1}h_{j1}, & h_{i1}h_{j2} + h_{i2}h_{j1}, & h_{i2}h_{j2}, & h_{i3}h_{j1} + h_{i1}h_{j3}, & h_{i3}h_{j2} + h_{i2}h_{j3}, & h_{i3}h_{j3} \end{bmatrix}^T$, then the equations B.9 and B.10 are rewritten as follows:

$$\begin{bmatrix} v_{12}^T \\ (v_{11}^T - v_{22}^T) \end{bmatrix}^T b = 0 \tag{B.15}$$

If a number of images n is used in the calibration process, then n of B.15 equations will be generated, therefore equation B.15 in general will be:

$$Vb = 0 \tag{B.16}$$

where, V is a matrix of dimensions $2n \times 6$. If $n = 1$, then two intrinsic parameters can be solved, and if $n = 2$, then the skewless coefficient will be zero $\gamma = 0$, and if $n \geq 3$ a unique solution for b matrix defined up to a scale factor.

The solution of B.16 is defined as the eigenvector of $V^T V$ associated with the smallest eigenvalue. Finally, once b is estimated, the intrinsic parameters can be

calculated as follows:

$$f_x = \sqrt{\frac{\lambda}{B_{11}}} \tag{B.17}$$

$$f_y = \sqrt{\frac{\lambda B_{11}}{(B_{11}B_{22} - B_{12}^2)}} \tag{B.18}$$

$$c_x = \frac{-B_{13}f_x^2}{\lambda} \tag{B.19}$$

$$c_y = \frac{B_{12}B_{13} - B_{11}B_{23}}{(B_{11}B_{22} - B_{12}^2)} \tag{B.20}$$

and consequently, the extrinsic parameters can be calculated using the equation B.8.

Bibliography

[1] Aarti, P. and Jimoh O, P. (2013). An Overview of Aerial Refueling Control Systems applied to UAVs. In *AFRICON, 2013*, pages 1–5. Ransleys.

[2] Abbott, E. and Powell, D. (1999). Land-vehicle navigation using gps. *Proceedings of the IEEE*, 87(1):145–162.

[3] Achtelik, M., Zhang, T., Kuhnlenz, K., and Buss, M. (2009). Visual tracking and control of a quadcopter using a stereo camera system and inertial sensors. In *Mechatronics and Automation, 2009. ICMA 2009. International Conference on*, pages 2863–2869.

[4] Adams, S. M. and Friedland, C. J. (2011). A survey of unmanned aerial vehicle (UAV) usage for imagery collection in disaster research and management. In *9th International Workshop on Remote Sensing for Disaster Response*.

[5] Agcayazi, M. T., Cawi, E., Jurgenson, A., Ghassemi, P., and Cook, G. (2016). ResQuad: Toward a semi-autonomous wilderness search and rescue unmanned aerial system. In *Unmanned Aircraft Systems (ICUAS), 2016 International Conference on*, pages 898–904. IEEE.

[6] Agrawal, M., Konolige, K., and Blas, M. R. (2008). Censure: Center surround extremas for realtime feature detection and matching. In *Computer Vision–ECCV 2008*, pages 102–115. Springer.

[7] Aguilar, W. G., Casaliglla, V. P., and Pólit, J. L. (2017). Obstacle avoidance for low-cost uavs. In *Semantic Computing (ICSC), 2017 IEEE 11th International Conference on*, pages 503–508. IEEE.

[8] Ahmad, A., Tahar, K. N., Udin, W. S., Hashim, K. A., Darwin, N., Hafis, M., Room, M., Hamid, N. F. A., Azhar, N. A. M., and Azmi, S. M. (2013). Digital aerial imagery of unmanned aerial vehicle for various applications. In *Control System, Computing and Engineering (ICCSCE), 2013 IEEE International Conference on*, pages 535–540. IEEE.

[9] Ahrens, S., Levine, D., Andrews, G., and How, J. P. (2009). Vision-based guidance and control of a hovering vehicle in unknown, GPS-denied environments. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 2643–2648. IEEE.

[10] Al Habsi, S., Shehada, M., Abdoon, M., Mashood, A., and Noura, H. (2015). Integration of a Vicon camera system for indoor flight of a Parrot AR Drone. In *Mechatronics and its Applications (ISMA), 2015 10th International Symposium on*, pages 1–6. IEEE.

[11] Al-Kaff, A., de la Escalera, A., and Armingol, José María, A. (2015). SIFT and SURF Performance Evaluation and the Effect of FREAK Descriptor in the Context of Visual Odometry for Unmanned Aerial Vehicles. In *International Conference on Computer Aided Systems Theory, EUROCAST*, pages 739–747. Springer.

[12] Al-Kaff, A., de la Escalera, A., and Armingol, J. M. (2017a). Indoor and outdoor navigational system for uavs based on monocular onboard camera. In *Advanced Concepts for Intelligent Vision Systems*. Springer.

[13] Al-Kaff, A., García, F., Martín, D., De La Escalera, A., and Armingol, J. M. (2017b). Obstacle detection and avoidance system based on monocular camera and size expansion algorithm for uavs. *Sensors*, 17(5):1061.

[14] Al-Kaff, A., Martín, D., García, F., de la Escalera, A., and Armingol, J. M. (2017c). Survey of computer vision algorithms and applications for unmanned aerial vehicles. *Expert Systems with Applications*.

[15] Al-Kaff, A., Meng, Q., Martín, D., de la Escalera, A., and Armingol, J. M. (2016). Monocular vision-based obstacle detection/avoidance for unmanned aerial vehicles. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 92–97. IEEE.

[16] Al-Kaff, A., Moreno, F. M., San José, L. J., García, F., Martín, D., de la Escalera, A., Nieva, A., and García, J. L. M. (2017d). Vbii-uav: Vision-based infrastructure inspection-uav. In *World Conference on Information Systems and Technologies, WorldCist'17*, pages 221–231. Springer.

[17] Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). Freak: Fast retina keypoint. In *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on*, pages 510–517. IEEE.

[18] Albaker, B. M. and Rahim, N. A. (2009). A survey of collision avoidance approaches for unmanned aerial vehicles. In *Technical Postgraduates (TECHPOS), 2009 International Conference for*, pages 1–7. IEEE.

[19] Altug, E., Ostrowski, J. P., and Mahony, R. (2002). Control of a quadrotor helicopter using visual feedback. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, pages 72–77. IEEE.

[20] Amini, A., Vaghefi, R. M., de la Garza, J. M., and Buehrer, R. M. (2014). Improving GPS-based vehicle positioning for intelligent transportation systems. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 1023–1029. IEEE.

[21] Amor-Martinez, A., Ruiz, A., Moreno-Noguer, F., and Sanfeliu, A. (2014). On-board real-time pose estimation for UAVs using deformable visual contour registration. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2595–2601. IEEE.

[22] Amorós, F., Paya, L., Valiente, D., Gil, A., and Reinoso, O. (2014). Visual odometry using the global-appearance of omnidirectional images. In *Informatics in Control, Automation and Robotics (ICINCO), 2014 11th International Conference on*, volume 2, pages 483–490. IEEE.

[23] Andrey, S. (2010). Tight Coupling of GPS and INS for Urban Navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 10(4):1731–1746.

[24] Angeli, A., Filliat, D., Doncieux, S., and Meyer, J.-A. (2006). 2d simultaneous localization and mapping for micro air vehicles. In *European Micro Aerial Vehicles (EMAV)*.

[25] Anthony, D., Elbaum, S., Lorenz, A., and Detweiler, C. (2014). On crop height estimation with UAVs. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 4805–4812. IEEE.

[26] Araar, O. and Aouf, N. (2014). Visual servoing of a Quadrotor UAV for autonomous power lines inspection. In *Control and Automation (MED), 2014 22nd Mediterranean Conference of*, pages 1418–1424. IEEE.

[27] Ariyur, K. B., Lommel, P., and Enns, D. F. (2005). Reactive inflight obstacle avoidance via radar feedback. In *American Control Conference, 2005. Proceedings of the 2005*, pages 2978–2982. IEEE.

[28] Babaei, A., Mortazavi, M., and Moradi, M. (2011). Classical and fuzzy-genetic autopilot design for unmanned aerial vehicles. *Applied Soft Computing*, 11(1):365–372.

[29] Bachrach, A., Prentice, S., He, R., Henry, P., Huang, A. S., Krainin, M., Maturana, D., Fox, D., and Roy, N. (2012). Estimation, planning, and mapping for autonomous flight using an rgb-d camera in gps-denied environments. *The International Journal of Robotics Research*, 31(11):1320–1343.

[30] Baek, J.-Y., Park, S.-H., Cho, B.-S., and Lee, M.-C. (2015). Position tracking system using single RGB-D Camera for evaluation of multi-rotor UAV control and self-localization. In *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1283–1288. IEEE.

[31] Bahat, M. F. and Filik, T. (2015). GPS-based antenna tracking and signal beamforming system for small UAV platform. In *Signal Processing and Communications Applications Conference (SIU), 2015 23th*, pages 1977–1980. IEEE.

[32] Bai, M., Wang, X., Yin, Y., and Xu, D. (2014). Aerial refueling drogue detection based on sliding-window object detector and hybrid features. In *Intelligent Control and Automation (WCICA), 2014 11th World Congress on*, pages 81–85. IEEE.

[33] Baik, K., Shin, J., Ji, S., Shon, W., and Park, S. (2011). A vision system for UAV position control. In *Aerospace Conference, 2011 IEEE*, pages 1–6. IEEE.

[34] Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117.

[35] Barczyk, M. and Lynch, A. F. (2012). Integration of a triaxial magnetometer into a helicopter uav gps-aided ins. *IEEE transactions on Aerospace and Electronic Systems*, 48(4):2947–2960.

[36] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer.

[37] Beard, R. W., Kingston, D., Quigley, M., Snyder, D., Christiansen, R., Johnson, W., McLain, T., and Goodrich, M. (2005). Autonomous vehicle technologies for small fixed-wing UAVs. *Journal of Aerospace Computing, Information, and Communication*, 2(1):92–108.

[38] Beard, R. W. and McLain, T. W. (2012). *Small unmanned aircraft theory and practice*. Princeton University Press.

[39] Beck, H., Lesueur, J., Charland-Arcand, G., Akhrif, O., Gagnı, S., Gagnon, F., and Couillard, D. (2016). Autonomous takeoff and landing of a quadcopter. In *Unmanned Aircraft Systems (ICUAS), 2016 International Conference on*, pages 475–484. IEEE.

[40] Benitez, W., Bogado, Y., Guerrero, A., and Arzamendia, M. (2016). Development of an uav prototype for visual inspection of aerial electrical lines. In *Embedded Systems (CASE), 2016 Seventh Argentine Conference on*, pages 7–12. IEEE.

[41] Bergerman, M., Amidi, O., Miller, J. R., Vallidis, N., and Dudek, T. (2007). Cascaded position and heading control of a robotic helicopter. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 135–140. IEEE.

[42] Beyeler, A., Zufferey, J.-C., and Floreano, D. (2007). 3d vision-based navigation for indoor microflyers. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 1336–1341. IEEE.

[43] Bills, C., Chen, J., and Saxena, A. (2011). Autonomous mav flight in indoor environments using single image perspective cues. In *Robotics and automation (ICRA), 2011 IEEE international conference on*, pages 5776–5783. IEEE.

[44] Bloesch, M., Omari, S., Hutter, M., and Siegwart, R. (2015). Robust visual inertial odometry using a direct EKF-based approach. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 298–304. IEEE.

[45] Blösch, M., Weiss, S., Scaramuzza, D., and Siegwart, R. (2010). Vision based mav navigation in unknown and unstructured environments. In *Robotics and automation (ICRA), 2010 IEEE international conference on*, pages 21–28. IEEE.

[46] Bonin-Font, F., Cosic, A., Negre, P. L., Solbach, M., and Oliver, G. (2015). Stereo SLAM for robust dense 3d reconstruction of underwater environments. In *OCEANS 2015-Genova*, pages 1–6. IEEE.

[47] Bošnak, M., Matko, D., and Blažič, S. (2012). Quadrocopter control using an onboard video system with off-board processing. *Robotics and Autonomous Systems*, 60(4):657–667.

[48] Bouguet, J.-Y. (2001). Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, 5(1-10):4.

[49] Bristeau, P.-J., Callou, F., Vissiere, D., and Petit, N. (2011). The navigation and control technology inside the ar. drone micro uav. *IFAC Proceedings Volumes*, 44(1):1477–1484.

[50] Broggi, A., Cattani, S., Patander, M., Sabbatelli, M., and Zani, P. (2013). A full-3d voxel-based dynamic obstacle detection for urban scenario using stereo vision. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 71–76. IEEE.

[51] Bry, A., Bachrach, A., and Roy, N. (2012). State estimation for aggressive flight in GPS-denied environments using onboard sensing. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1–8. IEEE.

[52] Burri, M., Nikolic, J., Hurzeler, C., Caprari, G., and Siegwart, R. (2012). Aerial service robots for visual inspection of thermal power plant boiler systems. In *Applied Robotics for the Power Industry (CARPI), 2012 2nd International Conference on*, pages 70–75. IEEE.

[53] Byrne, J. and Taylor, C. J. (2009). Expansion segmentation for visual collision detection and estimation. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 875–882. IEEE.

[54] Cabecinhas, D., Naldi, R., Marconi, L., Silvestre, C., and Cunha, R. (2010). Robust take-off and landing for a quadrotor vehicle. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1630–1635. IEEE.

[55] Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). Brief: Binary robust independent elementary features. In *Computer Vision–ECCV 2010*, pages 778–792. Springer.

[56] Campa, G., Napolitano, M. R., and Farvolini, M. L. (2009). Simulation Environment for Machine Vision Based Aerial Refueling for UAVs. *IEEE Transactions on Aerospace and Electronic Systems*, 45(1):138–151.

[57] Campoy, P., Correa, J. F., Mondragón, I., Martínez, C., Olivares, M., Mejías, L., and Artieda, J. (2009). Computer vision onboard uavs for civilian tasks. *Unmanned Aircraft Systems*, pages 105–135.

[58] Cao, W., Zhu, L., Han, J., Wang, T., and Du, Y. (2013). High voltage transmission line detection for uav based routing inspection. In *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 554–558. IEEE.

[59] Casau, P., Cabecinhas, D., and Silvestre, C. (2011). Autonomous transition flight for a vertical take-off and landing aircraft. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 3974–3979. IEEE.

[60] Çelik, K. and Somani, A. K. (2013). Monocular vision slam for indoor aerial vehicles. *Journal of electrical and computer engineering*, 2013:4–1573.

[61] Chakraborty, A. and Srinivasan, B. (2015). A novel stereo based obstacle avoidance system for unmanned aerial vehicles. *International Journal of Computer Applications*, 131(13):39–42.

[62] Chan, B., Guan, H., Jo, J., and Blumenstein, M. (2015). Towards uav-based bridge inspection systems: A review and an application perspective. *Structural Monitoring and Maintenance*, 2(3):283–300.

[63] Chavez, A. and Gustafson, D. (2009). Vision-based obstacle avoidance using SIFT features. In *Advances in Visual Computing*, pages 550–557. Springer.

[64] Chen, L., Jia, Q., and Zhang, X. (2010). Decoupling design of refueling boom by output feedback eigen structure assignment. *Computer Simulation*, 27(11):28–31.

[65] Cheng, Y., Maimone, M., and Matthies, L. (2005). Visual odometry on the Mars Exploration Rovers. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 903–910 Vol. 1.

[66] Cho, A., Kim, J., Lee, S., Choi, S., Lee, B., Kim, B., Park, N., Kim, D., and Kee, C. (2007). Fully automatic taxiing, takeoff and landing of a UAV using a single-antenna GPS receiver only. In *Control, Automation and Systems, 2007. ICCAS'07. International Conference on*, pages 821–825. IEEE.

[67] Cho, O.-h., Ban, K.-j., and Kim, E.-k. (2014). Stabilized UAV flight system design for structure safety inspection. In *Advanced Communication Technology (ICACT), 2014 16th International Conference on*, pages 1312–1316. Citeseer.

[68] Choi, S. and Kim, E. (2015a). Image Acquisition System for Construction Inspection Based on Small Unmanned Aerial Vehicle. In *Advanced Multimedia and Ubiquitous Engineering*, pages 273–280. Springer.

[69] Choi, S.-s. and Kim, E.-k. (2015b). Building crack inspection using small uav. In *2015 17th International Conference on Advanced Communication Technology (ICACT)*, pages 235–238. IEEE.

[70] Chunhui, Z., Rongzhi, W., Tianwu, Z., and Quan, P. (2014). Visual odometry and scene matching integrated navigation system in UAV. In *Information Fusion (FUSION), 2014 17th International Conference on*, pages 1–6. IEEE.

[71] Chunhui, Z., Shan, J., Haifeng, Z., Kaixuan, Z., Yang, L., and Qingchun, Z. (2016). Aerial target detection and avoidance for uav based on stereo vision. In *Guidance, Navigation and Control Conference (CGNCC), 2016 IEEE Chinese*, pages 1670–1675. IEEE.

[72] Cocchioni, F., Mancini, A., and Longhi, S. (2014). Autonomous navigation, landing and recharge of a quadrotor using artificial vision. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pages 418–429. IEEE.

[73] Cornall, T., Egan, G., and Price, A. (2006). Aircraft attitude estimation from horizon video. *Electronics Letters*, 42(13):744–745.

[74] Cornall, T. D. and Egan, G. (2004). Measuring horizon angle from video on a small unmanned air vehicle. In *2nd international conference on autonomous robots and agents*, volume 6, page 7.

[75] Costa, B. S. J., Greati, V. R., Ribeiro, V. C. T., da Silva, C. S., and Vieira, I. F. (2015). A visual protocol for autonomous landing of unmanned aerial vehicles based on fuzzy matching and evolving clustering. In *Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on*, pages 1–6. IEEE.

[76] Csorba, M. (1997). *Simultaneous localisation and map building*. PhD thesis, University of Oxford.

[77] Cui, H., Lin, Z., and Zhang, J. (2007). Research on Low Altitude Image Acquisition System. In *Computer And Computing Technologies In Agriculture, Volume I*, pages 95–102. Springer.

[78] Cui, J. Q., Lai, S., Dong, X., Liu, P., Chen, B. M., and Lee, T. H. (2014). Autonomous navigation of UAV in forest. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pages 726–733. IEEE.

[79] Cusack, B. and Khaleghparast, R. (2015). Evaluating small drone surveillance capabilities to enhance traffic conformance intelligence.

[80] Daibing, Z., Xun, W., and Weiwei, K. (2012). Autonomous control of running takeoff and landing for a fixed-wing unmanned aerial vehicle. In *Control Automation Robotics & Vision (ICARCV), 2012 12th International Conference on*, pages 990–994. IEEE.

[81] Dalamagkidis, K., Valavanis, K., and Piegl, L. A. (2011). *On integrating unmanned aircraft systems into the national airspace system: issues, challenges, operational restrictions, certification, and recommendations*, volume 54. Springer Science & Business Media.

[82] Davison, A. J., Reid, I. D., Molton, N. D., and Stasse, O. (2007). Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6).

[83] de Araujo, V., Almeida, A. P. G., Miranda, C. T., and de Barros Vidal, F. (2014). A parallel hierarchical finite state machine approach to UAV control for search and rescue tasks. In *Informatics in Control, Automation and Robotics (ICINCO), 2014 11th International Conference on*, volume 1, pages 410–415. IEEE.

[84] De Croon, G., De Weerdt, E., De Wagter, C., Remes, B., and Ruijsink, R. (2012). The appearance variation cue for obstacle avoidance. *IEEE Transactions on Robotics*, 28(2):529–534.

[85] Deng, Y., Xian, N., and Duan, H. (2016). A binocular vision-based measuring system for UAVs autonomous aerial refueling. In *Control and Automation (ICCA), 2016 12th IEEE International Conference on*, pages 221–226. IEEE.

[86] Domínguez, S., Zalama, E., García-Bermejo, J. G., Worst, R., and Behnke, S. (2013). Fast 6d Odometry Based on Visual Features and Depth. In *Frontiers of Intelligent Autonomous Systems*, pages 5–16. Springer.

[87] Du, S. and Tu, C. (2011). Power line inspection using segment measurement based on HT butterfly. In *Signal Processing, Communications and Computing (ICSPCC), 2011 IEEE International Conference on*, pages 1–4. IEEE.

[88] Duan, H. and Zhang, Q. (2015). Visual measurement in simulation environment for vision-based uav autonomous aerial refueling. *IEEE Transactions on Instrumentation and Measurement*, 64(9):2468–2480.

[89] Dunbabin, M., Corke, P., and Buskey, G. (2004). Low-cost vision-based AUV guidance system for reef navigation. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, pages 7–12. IEEE.

[90] Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110.

[91] Dusha, D., Boles, W. W., and Walker, R. (2007). Fixed-wing attitude estimation using computer vision based horizon detection. In *12th Australian International Aerospace Congress*, pages 1–19.

[92] Eberli, D., Scaramuzza, D., Weiss, S., and Siegwart, R. (2011). Vision based position control for mavs using one single circular landmark. *Journal of Intelligent & Robotic Systems*, 61(1-4):495–512.

[93] Elkaim, G. H., Lie, F. A. P., and Gebre-Egziabher, D. (2015). Principles of Guidance, Navigation, and Control of UAVs. In *Handbook of Unmanned Aerial Vehicles*, pages 347–380. Springer.

[94] Engel, J., Koltun, V., and Cremers, D. (2017). Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[95] Engel, J., Schöps, T., and Cremers, D. (2014). LSD-SLAM: Large-scale direct monocular SLAM. In *Computer Vision–ECCV 2014*, pages 834–849. Springer.

[96] Engel, J., Usenko, V., and Cremers, D. (2016). A photometrically calibrated benchmark for monocular visual odometry. *arXiv preprint arXiv:1607.02555*.

[97] Erdos, D., Erdos, A., and Watkins, S. E. (2013). An experimental uav system for search and rescue challenge. *IEEE Aerospace and Electronic Systems Magazine*, 28(5):32–37.

[98] Eresen, A., İmamoğlu, N., and Efe, M. Ö. (2012). Autonomous quadrotor flight with vision-based obstacle avoidance in virtual environment. *Expert Systems with Applications*, 39(1):894–905.

[99] Eschmann, C., Kuo, C., Kuo, C., and Boller, C. (2012). Unmanned aircraft systems for remote building inspection and monitoring. In *6th European workshop on structural health monitoring*, pages 1–8.

[100] Fabian, J. and Clayton, G. M. (2014). Error analysis for visual odometry on indoor, wheeled mobile robots with 3-d sensors. *IEEE/ASME Transactions on Mechatronics*, 19(6):1896–1906.

[101] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

[102] Fraundorfer, F., Heng, L., Honegger, D., Lee, G. H., Meier, L., Tanskanen, P., and Pollefeys, M. (2012). Vision-based autonomous mapping and exploration using a quadrotor mav. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4557–4564. IEEE.

[103] Freed, M., Harris, R., and Shafto, M. (2004). Measuring Autonomous UAV Surveillance Performance. *Proceedings of PerMIS'04*.

[104] Fu, C., Carrio, A., and Campoy, P. (2015). Efficient visual odometry and mapping for Unmanned Aerial Vehicle using ARM-based stereo vision preprocessing system. In *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, pages 957–962. IEEE.

[105] Fu, C., Olivares-Mendez, M. A., Suarez-Fernandez, R., and Campoy, P. (2014). Monocular visual-inertial slam-based collision avoidance strategy for fail-safe uav using fuzzy logic controllers. *Journal of Intelligent & Robotic Systems*, 73(1-4):513–533.

[106] Fu, M., Zhang, K., Yi, Y., and Shi, C. (2016). Autonomous landing of a quadrotor on an UGV. In *Mechatronics and Automation (ICMA), 2016 IEEE International Conference on*, pages 988–993. IEEE.

[107] Gageik, N., Benz, P., and Montenegro, S. (2015). Obstacle detection and collision avoidance for a uav with complementary low-cost sensors. *IEEE Access*, 3:599–609.

[108] Gao, C., Zhao, G., and Lu, J. (2014). In-flight misalignment attitude estimation for UAV based on GPS and UKF/MPF filter. In *Guidance, Navigation and Control Conference (CGNCC), 2014 IEEE Chinese*, pages 452–457. IEEE.

[109] Gao, Y., Ai, X., Rarity, J., and Dahnoun, N. (2011). Obstacle detection with 3d camera using uv-disparity. In *Systems, Signal Processing and their Applications (WOSSPA), 2011 7th International Workshop on*, pages 239–242. IEEE.

[110] Gautam, A., Sujit, P. B., and Saripalli, S. (2014). A survey of autonomous landing techniques for UAVs. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pages 1210–1218. IEEE.

[111] Geng, L., Zhang, Y. F., Wang, J. J., Fuh, J. Y., and Teo, S. H. (2013). Mission planning of autonomous UAVs for urban surveillance with evolutionary algorithms. In *Control and Automation (ICCA), 2013 10th IEEE International Conference on*, pages 828–833. IEEE.

[112] Geng, L., Zhang, Y. F., Wang, P. F., Wang, J. J., Fuh, J. Y., and Teo, S. H. (2014). UAV surveillance mission planning with gimbaled sensors. In *Control & Automation (ICCA), 11th IEEE International Conference on*, pages 320–325. IEEE.

[113] Ghommam, J. and Saad, M. (2017). Autonomous landing of a quadrotor on a moving platform. *IEEE Transactions on Aerospace and Electronic Systems*.

[114] Gibson, J. J. (2014). *The ecological approach to visual perception: classic edition*. Psychology Press.

[115] Golightly, I. and Jones, D. (2005). Visual control of an unmanned aerial vehicle for power line inspection. In *Advanced Robotics, 2005. ICAR'05. Proceedings., 12th International Conference on*, pages 288–295. IEEE.

[116] Gomes, L. L., Leal, L. P., Oliveira, T. R., da Cunha, J. P. V. S., and Revoredo, T. C. (2016). Unmanned quadcopter control using a motion capture system. *IEEE Latin America Transactions*, 14(8):3606–3613.

[117] Gopalakrishnan, K., Khaitan, S. K., and Kalogirou, S. (2011). *Soft computing in green and renewable energy systems*, volume 269. Springer.

[118] Gotovac, D., Gotovac, S., and Papić, V. (2016). Mapping aerial images from uav. In *Computer and Energy Science (SpliTech), International Multidisciplinary Conference on*, pages 1–6. IEEE.

[119] Govindaraju, V., Leng, G., and Qian, Z. (2014). Visibility-based UAV path planning for surveillance in cluttered environments. In *Safety, Security, and Rescue Robotics (SSRR), 2014 IEEE International Symposium on*, pages 1–6. IEEE.

[120] Grabe, V., Bülthoff, H. H., and Giordano, P. R. (2012). Robust optical-flow based self-motion estimation for a quadrotor uav. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2153–2159. IEEE.

[121] Green, W. and Oh, P. (2008). Optic-Flow-Based Collision Avoidance. *IEEE Robotics Automation Magazine*, 15(1):96–103.

[122] Grelsson, B. (2014). *Global Pose Estimation from Aerial Images: Registration with Elevation Models*. PhD thesis, Linköping University Electronic Press.

[123] Grzonka, S., Grisetti, G., and Burgard, W. (2012). A fully autonomous indoor quadrotor. *IEEE Transactions on Robotics*, 28(1):90–100.

[124] Guizilini, V. and Ramos, F. (2011). Visual odometry learning for unmanned aerial vehicles. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 6213–6220. IEEE.

[125] Hackney, C. and Clayton, A. (2015). 2.1. 7. unmanned aerial vehicles (uavs) and their application in geomorphic mapping.

[126] Hammer, A., Dumoulin, J., Vozel, B., and Chehdi, K. (2007). Deblurring of UAV aerial images for civil structures inspections using Mumford-Shah/Total variation regularisation. In *Image and Signal Processing and Analysis, 2007. ISPA 2007. 5th International Symposium on*, pages 262–267. IEEE.

[127] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer.

[128] Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.

[129] Hartmann, K. and Giles, K. (2016). Uav exploitation: A new domain for cyber power. In *Cyber Conflict (CyCon), 2016 8th International Conference on*, pages 205–221. NATO CCD COE.

[130] Heintz, F., Rudol, P., and Doherty, P. (2007). From images to traffic behavior-a uav tracking and monitoring application. In *Information Fusion, 2007 10th International Conference on*, pages 1–8. IEEE.

[131] Herissé, B., Hamel, T., Mahony, R., and Russotto, F.-X. (2012). Landing a vtol unmanned aerial vehicle on a moving platform using optical flow. *IEEE Transactions on Robotics*, 28(1):77–89.

[132] Høglund, S. (2014). *Autonomous inspection of wind turbines and buildings using an UAV*. PhD thesis, Institutt for teknisk kybernetikk.

[133] Horn, B. K. and Schunck, B. G. (1992). Determining optical flow. In *Shape recovery*, pages 389–407. Jones and Bartlett Publishers, Inc.

[134] Hou, J., Zhang, Q., Zhang, Y., Zhu, K., Lv, Y., and Yu, C. (2016). Low altitude sense and avoid for muav based on stereo vision. In *Control Conference (CCC), 2016 35th Chinese*, pages 5579–5584. TCCT.

[135] Hrabar, S. (2008). 3d path planning and stereo-based obstacle avoidance for rotorcraft uavs. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 807–814. IEEE.

[136] Hrabar, S., Sukhatme, G. S., Corke, P., Usher, K., and Roberts, J. (2005). Combined optic-flow and stereo-based navigation of urban canyons for a uav. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3309–3316. IEEE.

[137] Huan, D., Guoliang, F., and Jianqiang, Y. (2015). Autonomous landing for unmanned seaplanes based on active disturbance rejection control. In *Control Conference (CCC), 2015 34th Chinese*, pages 5663–5668. IEEE.

[138] Hui, C., Yousheng, C., Xiaokun, L., and Shing, W. W. (2013). Autonomous takeoff, tracking and landing of a UAV on a moving UGV using onboard monocular vision. In *Control Conference (CCC), 2013 32nd Chinese*, pages 5895–5901. IEEE.

[139] Hui, Y., Xhiping, C., Shanjia, X., and Shisong, W. (1998). An unmanned air vehicle (uav) gps location and navigation system. In *Microwave and Millimeter Wave Technology Proceedings, 1998. ICMMT'98. 1998 International Conference on*, pages 472–475. IEEE.

[140] Hussein, A., Al-Kaff, A., de la Escalera, A., and Armingol, J. M. (2015). Autonomous indoor navigation of low-cost quadcopters. In *Service Operations And Logistics, And Informatics (SOLI), 2015 IEEE International Conference on*, pages 133–138. IEEE.

[141] Hussein, A., Mostafa, H., Badrel-din, M., Sultan, O., and Khamis, A. (2012). Metaheuristic optimization approach to mobile robot path planning. In *Engineering and Technology (ICET), 2012 International Conference on*, pages 1–6. IEEE.

[142] Isaacs, J. T., Magee, C., Subbaraman, A., Quitin, F., Fregene, K., Teel, A. R., Madhow, U., and Hespanha, J. P. (2014). GPS-optimal micro air vehicle navigation in degraded environments. In *American Control Conference (ACC), 2014*, pages 1864–1871. IEEE.

[143] Jeong, C. Y. and Choi, S. (2016). A comparison of keypoint detectors in the context of pedestrian counting. In *Information and Communication Technology Convergence (ICTC), 2016 International Conference on*, pages 1179–1181. IEEE.

[144] Jeong, J., Mulligan, J., and Correll, N. (2013). Trinocular visual odometry for divergent views with minimal overlap. In *Robot Vision (WORV), 2013 IEEE Workshop on*, pages 229–236. IEEE.

[145] Jian, L. and Xiao-min, L. (2011). Vision-based navigation and obstacle detection for uav. In *Electronics, Communications and Control (ICECC), 2011 International Conference on*, pages 1771–1774. IEEE.

[146] Jiong, Y., Lei, Z., Jiangping, D., Rong, S., and Jianyu, W. (2010). GPS/SINS/BARO Integrated Navigation System for UAV. In *Information Technology and Applications (IFITA), 2010 International Forum on*, volume 3, pages 19–25. IEEE.

[147] Jung, Y., Bang, H., and Lee, D. (2015). Robust marker tracking algorithm for precise UAV vision-based autonomous landing. In *Control, Automation and Systems (ICCAS), 2015 15th International Conference on*, pages 443–446. IEEE.

[148] Kada, B. and Ghazzawi, Y. (2011). Robust pid controller design for an uav flight control system. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 2, pages 19–21.

[149] Kaiser, K., Gans, N., and Dixon, W. (2006). Position and orientation of an aerial vehicle through chained, vision-based pose reconstruction. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 6719.

[150] Kamel, B., Santana, M. C. S., and De Almeida, T. C. (2010). Position estimation of autonomous aerial navigation based on hough transform and harris corners detection. *Recent Researches in Circuits, Systems, Electronics, Control & Signal Processing*, pages 148–154.

[151] Kanistras, K., Martins, G., Rutherford, M. J., and Valavanis, K. P. (2015). Survey of unmanned aerial vehicles (uavs) for traffic monitoring. In *Handbook of Unmanned Aerial Vehicles*, pages 2643–2666. Springer.

[152] Kaplan, E. and Hegarty, C. (2005). *Understanding GPS: principles and applications*. Artech house.

[153] Ke, R., Kim, S., Li, Z., and Wang, Y. (2015). Motion-vector clustering for traffic speed detection from UAV video. In *IEEE First International of Smart Cities Conference (ISC2)*, pages 1–5. IEEE.

[154] Keke, L., Qing, L., and Nong, C. (2014). An autonomous carrier landing system design and simulation for unmanned aerial vehicle. In *Guidance, Navigation and Control Conference (CGNCC), 2014 IEEE Chinese*, pages 1352–1356. IEEE.

[155] Kendoul, F. (2012). Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics*, 29(2):315–378.

[156] Kerl, C. (2012). *Odometry from RGB-D Cameras for Autonomous Quadrocopters*. Master Thesis, Technischen Universitat Munchen, Munchen, Germany.

[157] Kerl, C., Sturm, J., and Cremers, D. (2013). Dense visual SLAM for RGB-D cameras. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 2100–2106. IEEE.

[158] Killpack, M., Deyle, T., Anderson, C., and Kemp, C. C. (2010). Visual odometry and control for an omnidirectional mobile robot with a downward-facing camera. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 139–146. IEEE.

[159] Kim, H. J., Kim, M., Lim, H., Park, C., Yoon, S., Lee, D., Choi, H., Oh, G., Park, J., and Kim, Y. (2013). Fully autonomous vision-based net-recovery landing system for a fixed-wing uav. *IEEE/ASME Transactions On Mechatronics*, 18(4):1320–1333.

[160] Kim, J. (2004). *Autonomous navigation for airborne applications*. PhD thesis, Department of Aerospace, Mechanical and Mechatronic Engineering, Graduate School of Engineering, University of Sydney.

[161] Kim, J. and Do, Y. (2012). Moving obstacle avoidance of a mobile robot using a single camera. *Procedia Engineering*, 41:911–916.

[162] Kim, S.-W., Gwon, G.-P., Choi, S.-T., Kang, S.-N., Shin, M.-O., Yoo, I.-S., Lee, E.-D., Frazzoli, E., and Seo, S.-W. (2012). Multiple vehicle driving control for traffic flow efficiency. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 462–468. IEEE.

[163] Kim, Z. (2006). Geometry of vanishing points and its application to external calibration and realtime pose estimation. *Institute of Transportation Studies*.

[164] Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE.

[165] Kneip, L., Chli, M., Siegwart, R., et al. (2011). Robust real-time visual odometry with a single camera and an imu.

[166] Kong, W., Zhang, D., and Zhang, J. (2015). A ground-based multi-sensor system for autonomous landing of a fixed wing UAV. In *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1303–1310. IEEE.

[167] Kong, W., Zhou, D., Zhang, Y., Zhang, D., Wang, X., Zhao, B., Yan, C., Shen, L., and Zhang, J. (2014). A ground-based optical system for autonomous landing of a fixed wing UAV. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4797–4804. IEEE.

[168] Kothari, M., Postlethwaite, I., and Gu, D.-W. (2009). Multi-UAV path planning in obstacle rich environments using rapidly-exploring random trees. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 3069–3074. IEEE.

[169] Krajník, T., Nitsche, M., Pedre, S., Přeučil, L., and Mejail, M. E. (2012). A simple visual navigation system for an uav. In *Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on*, pages 34–39. IEEE.

[170] Krajník, T., Vonásek, V., Fi\vser, D., and Faigl, J. (2011). AR-drone as a platform for robotic research and education. In *Research and Education in Robotics-EUROBOT 2011*, pages 172–186. Springer.

[171] Kruijff, G.-J. M., Tretyakov, V., Linder, T., Pirri, F., Gianni, M., Papadakis, P., Pizzoli, M., Sinha, A., Pianese, E., Corrao, S., and others (2012). Rescue robots at earthquake-hit mirandola, italy: a field report. In *Safety, Security, and Rescue Robotics (SSRR), 2012 IEEE International Symposium on*, pages 1–8. IEEE.

[172] Kunz, C. and Singh, H. (2010). Stereo self-calibration for seafloor mapping using AUVs. In *Autonomous Underwater Vehicles (AUV), 2010 IEEE/OES*, pages 1–7. IEEE.

[173] Kurnaz, S., Cetin, O., and Kaynak, O. (2010). Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles. *Expert Systems with Applications*, 37(2):1229–1234.

[174] Lange, S., Sünderhauf, N., Neubert, P., Drews, S., and Protzel, P. (2012). Autonomous corridor flight of a UAV using a low-cost and light-weight RGB-D camera. In *Advances in Autonomous Mini Robots*, pages 183–192. Springer.

[175] Lee, D., Ryan, T., and Kim, H. J. (2012). Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 971–976. IEEE.

[176] Lee, I.-U., Li, H., Hoang, N.-M., and Lee, J.-M. (2014a). Navigation system development of the Underwater Vehicles using the GPS/INS sensor fusion. In *Control, Automation and Systems (ICCAS), 2014 14th International Conference on*, pages 610–612. IEEE.

[177] Lee, J.-O., Lee, K.-H., Park, S.-H., Im, S.-G., and Park, J. (2011). Obstacle avoidance for small uavs using monocular vision. *Aircraft Engineering and Aerospace Technology*, 83(6):397–406.

[178] Lee, M.-F. R., Su, S.-F., Yeah, J.-W. E., Huang, H.-M., and Chen, J. (2014b). Autonomous landing system for aerial mobile robot cooperation. In *Soft Computing and Intelligent Systems (SCIS), 2014 Joint 7th International Conference on and Advanced Intelligent Systems (ISIS), 15th International Symposium on*, pages 1306–1311. IEEE.

[179] Lenz, I., Gemici, M., and Saxena, A. (2012). Low-power parallel algorithms for single image based obstacle avoidance in aerial robots. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 772–779. IEEE.

[180] Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). BRISK: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555. IEEE.

[181] Li, B., Mu, C., and Wu, B. (2012a). A survey of vision based autonomous aerial refueling for unmanned aerial vehicles. In *Intelligent Control and Information Processing (ICICIP), 2012 Third International Conference on*, pages 1–6. IEEE.

[182] Li, K., Zhang, K., and Chen, B. M. (2016). On-board visual odometry and autonomous control of a quadrotor micro aerial vehicle. In *Control and Automation (ICCA), 2016 12th IEEE International Conference on*, pages 68–73. IEEE.

[183] Li, X., Aouf, N., and Nemra, A. (2012b). Estimation analysis in VSLAM for UAV application. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on*, pages 365–370. IEEE.

[184] Li, X. and Yang, L. (2012). Design and Implementation of UAV Intelligent Aerial Photography System. In *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference on*, volume 2, pages 200–203. IEEE.

[185] Li, Z., Jia, J., Cheng, M., and Cui, Z. (2014). Solving Path Planning of UAV Based on Modified Multi-Population Differential Evolution Algorithm. In *International Symposium on Neural Networks*, pages 602–610. Springer.

[186] Lilien, L. T., Othmane, L. b., Angin, P., Bhargava, B., Salih, R. M., and DeCarlo, A. (2015). Impact of Initial Target Position on Performance of UAV Surveillance Using Opportunistic Resource Utilization Networks. In *Reliable Distributed Systems Workshop (SRDSW), 2015 IEEE 34th Symposium on*, pages 57–61. IEEE.

[187] Lim, H., Lee, H., and Kim, H. J. (2012). Onboard flight control of a micro quadrotor using single strapdown optical flow sensor. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 495–500. IEEE.

[188] Limnaios, G. and Tsourveloudis, N. (2012). Fuzzy Logic Controller for a Mini Coaxial Indoor Helicopter. *Journal of Intelligent & Robotic Systems*, 65(1-4):187–201.

[189] Lin, C.-H., Jiang, S.-Y., Pu, Y.-J., and Song, K.-T. (2010). Robust ground plane detection for obstacle avoidance of mobile robots using a monocular camera. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3706–3711. IEEE.

[190] Lin, F., Lum, K.-Y., Chen, B. M., and Lee, T. H. (2009). Development of a vision-based ground target detection and tracking system for a small unmanned helicopter. *Science in China Series F: Information Sciences*, 52(11):2201–2215.

[191] Lin, Y. and Saripalli, S. (2012). Road detection from aerial imagery. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3588–3593. IEEE.

[192] Lindsten, F., Callmer, J., Ohlsson, H., Törnqvist, D., Schön, T. B., and Gustafsson, F. (2010). Geo-referencing for UAV navigation using environmental classification. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1420–1425. IEEE.

[193] Liu, Y.-c. and Dai, Q.-h. (2010). A survey of computer vision applied in aerial robotic vehicles. In *Optics Photonics and Energy Engineering (OPEE), 2010 International Conference on*, volume 1, pages 277–280. IEEE.

[194] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

[195] Lucas, B. D. and Kanade, T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 674–679, San Francisco, CA, USA. Vancouver.

[196] Lukashevich, P., Belotserkovsky, A., and Nedzved, A. (2015). The new approach for reliable uav navigation based on onboard camera image processing. In *Information and Digital Technologies (IDT), 2015 International Conference on*, pages 230–234. IEEE.

[197] Luo, D., Wang, F., Wang, B., and Chen, B. M. (2012). Implementation of obstacle avoidance technique for indoor coaxial rotorcraft with Scanning Laser Range Finder. In *Control Conference (CCC), 2012 31st Chinese*, pages 5135–5140. IEEE.

[198] Lyu, Y., Pan, Q., Zhang, Y., Zhao, C., Zhu, H., Tang, T., and Liu, L. (2015). Simultaneously multi-UAV mapping and control with visual servoing. In *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, pages 125–131. IEEE.

[199] Ma, L., Li, M., Tong, L., Wang, Y., and Cheng, L. (2013). Using unmanned aerial vehicle for remote sensing application. In *Geoinformatics (GEOINFORMATICS), 2013 21st International Conference on*, pages 1–5. IEEE.

[200] Ma, Z., Hu, T., Shen, L., Kong, W., and Zhao, B. (2015). A detection and relative direction estimation method for UAV in sense-and-avoid. In *Information and Automation, 2015 IEEE International Conference on*, pages 2677–2682. IEEE.

[201] Madison, R., Andrews, G., DeBitetto, P., Rasmussen, S., and Bottkol, M. (2007). Vision-aided navigation for small uavs in gps-challenged environments. In *AIAA Infotech@ Aerospace 2007 Conference and Exhibit*, page 2986.

[202] Magree, D., Mooney, J. G., and Johnson, E. N. (2013). Monocular visual mapping for obstacle avoidance on uavs. In *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*, pages 471–479. IEEE.

[203] Maimone, M., Cheng, Y., and Matthies, L. (2007). Two years of Visual Odometry on the Mars Exploration Rovers. *Journal of Field Robotics*, 24(3):169–186.

[204] Majumder, S., Shankar, R., and Prasad, M. S. (2015). Obstacle size and proximity detection using stereo images for agile aerial robots. In *Signal Processing and Integrated Networks (SPIN), 2015 2nd International Conference on*, pages 437–442. IEEE.

[205] Mamdani, E. (1976a). Advances in the linguistic synthesis of fuzzy controllers. *International Journal of Man-Machine Studies*, 8(6):669–678.

[206] Mamdani, E. H. (1976b). Application of fuzzy logic to approximate reasoning using linguistic synthesis. In *Proceedings of the sixth international symposium on Multiple-valued logic*, pages 196–202. IEEE Computer Society Press.

[207] Mamdani, E. H. and Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies*, 7(1):1–13.

[208] Mammarella, M., Campa, G., Napolitano, M. R., and Fravolini, M. L. (2010). Comparison of point matching algorithms for the UAV aerial refueling problem. *Machine Vision and Applications*, 21(3):241–251.

[209] Mart, J., Marquez, F., Garcia, E. O., Mu, A., Mayol-Cuevas, W., and others (2015). On combining wearable sensors and visual SLAM for remote controlling of low-cost micro aerial vehicles. In *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pages 232–240. IEEE.

[210] Martin, G. T. (2012). Modelling and Control of the Parrot AR.Drone. *The UNSW Canberra at ADFA Journal of Undergraduate Engineering Research*, 5(1).

[211] Martínez Luna, C. V. (2013). *Visual Tracking, Pose Estimation, and Control for Aerial Vehicles*. PhD thesis, Universidad Politecnica de Madrid.

[212] Martínez, C., Campoy, P., Mondragón, I., and Olivares Mendez, M. A. (2009). Trinocular ground system to control UAVs. In *2009 IEEE-RSJ INTERNATIONAL CONFERENCE ON INTILIGENT ROBOTS AND SYSTEMS*, pages 3361–3367. Ieee.

[213] Martínez, C., Richardson, T., and Campoy, P. (2013). Towards Autonomous Air-to-Air Refuelling for UAVs using visual information. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5756–5762. IEEE.

[214] Mati, R., Pollini, L., Lunghi, A., Innocenti, M., and Campa, G. (2006). Vision-based autonomous probe and drogue aerial refueling. In *2006 14th Mediterranean Conference on Control and Automation*, pages 1–6. IEEE.

[215] Mcfadyen, A. and Mejias, L. (2016). A survey of autonomous vision-based See and Avoid for Unmanned Aircraft Systems. *Progress in Aerospace Sciences*, 80:1–17.

[216] Mehmood, S., Choudhry, A. J., Anwar, H., Mahmood, S., and Khan, A. A. (2016). Stereo-vision based autonomous underwater navigation??? The platform SARSTION. In *2016 13th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pages 554–559. IEEE.

[217] Meldrum, D. T. and Haddrell, T. (1994). GPS in autonomous underwater vehicles. In *Electronic Engineering in Oceanography, 1994., Sixth International Conference on*, pages 11–17. IET.

[218] Mellinger, D., Michael, N., and Kumar, V. (2012). Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors. *The International Journal of Robotics Research*, 31(5):664–674.

[219] Meng, L., de Silva, C. W., and Zhang, J. (2014). 3d visual SLAM for an assistive robot in indoor environments using RGB-D cameras. In *Computer Science & Education (ICCSE), 2014 9th International Conference on*, pages 32–37. IEEE.

[220] Merrell, P. C., Lee, D.-J., and Beard, R. W. (2004). Obstacle avoidance for unmanned air vehicles using optical flow probability distributions. In *Optics East*, pages 13–22. International Society for Optics and Photonics.

[221] Metni, N. and Hamel, T. (2007). A UAV for bridge inspection: Visual servoing control law with orientation limits. *Automation in construction*, 17(1):3–10.

[222] Michael, N., Fink, J., and Kumar, V. (2011). Cooperative manipulation and transportation with aerial robots. *Autonomous Robots*, 30(1):73–86.

[223] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Van Gool, L. (2005). A Comparison of Affine Region Detectors. *International journal of computer vision*, 65(1-2):43–72.

[224] Milford, M. J., Schill, F., Corke, P., Mahony, R., and Wyeth, G. (2011). Aerial SLAM with a single camera using visual expectation. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2506–2512. IEEE.

[225] Milford, M. J., Wyeth, G. F., and Rasser, D. F. (2004). RatSLAM: a hippocampal model for simultaneous localization and mapping. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, pages 403–408. IEEE.

[226] Minh, L. D. and Ha, C. (2010). Modeling and control of quadrotor MAV using vision-based measurement. In *Strategic Technology (IFOST), 2010 International Forum on*, pages 70–75. IEEE.

[227] More, V., Kumar, H., Kaingade, S., Gaidhani, P., and Gupta, N. (2015). Visual odometry using optic flow for Unmanned Aerial Vehicles. In *Cognitive Computing and Information Processing (CCIP), 2015 International Conference on*, pages 1–6. IEEE.

[228] Mori, T. and Scherer, S. (2013). First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1750–1757. IEEE.

[229] Mouats, T., Aouf, N., Chermak, L., and Richardson, M. A. (2015). Thermal Stereo Odometry for UAVs. *IEEE Sensors Journal*, 15(11):6335–6347.

[230] Mourikis, A. I., Trawny, N., Roumeliotis, S. I., Johnson, A. E., Ansar, A., and Matthies, L. (2009). Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing. *IEEE Transactions on Robotics*, 25(2):264–280.

[231] Muskardin, T., Balmer, G., Wlach, S., Kondak, K., Laiacker, M., and Ollero, A. (2016). Landing of a fixed-wing UAV on a mobile ground vehicle. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1237–1242. IEEE.

[232] Na, I., Han, S. H., and Jeong, H. (2011). Stereo-based road obstacle detection and tracking. In *2011 13th International Conference on Advanced Communication Technology (ICACT)*, pages 1181–1184.

[233] Naidoo, Y., Stopforth, R., and Bright, G. (2011). Development of an uav for search & rescue applications. In *AFRICON, 2011*, pages 1–6. IEEE.

[234] Nakanishi, H., Kanata, S., and Sawaragi, T. (2011). Measurement model of barometer in ground effect of unmanned helicopter and its application to estimate terrain clearance. In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pages 232–237. IEEE.

[235] Nakanishi, H., Kanata, S., and Sawaragi, T. (2012). GPS-INS-BARO hybrid navigation system taking into account ground effect for autonomous unmanned helicopter. In *Safety, Security, and Rescue Robotics (SSRR), 2012 IEEE International Symposium on*, pages 1–6. IEEE.

[236] Navia, J., Mondragon, I., Patino, D., and Colorado, J. (2016). Multispectral mapping in agriculture: Terrain mosaic using an autonomous quadcopter UAV. In *Unmanned Aircraft Systems (ICUAS), 2016 International Conference on*, pages 1351–1358. IEEE.

[237] Neff, A. E., Lee, D., Chitrakaran, V. K., Dawson, D. M., and Burg, T. C. (2007). Velocity control for a quad-rotor uav fly-by-camera interface. In *SoutheastCon, 2007. Proceedings. IEEE*, pages 273–278. IEEE.

[238] Nikolic, J., Burri, M., Rehder, J., Leutenegger, S., Huerzeler, C., and Siegwart, R. (2013). A UAV system for inspection of industrial facilities. In *Aerospace Conference, 2013 IEEE*, pages 1–8. IEEE.

[239] Nikolic, J., Rehder, J., Burri, M., Gohl, P., Leutenegger, S., Furgale, P. T., and Siegwart, R. (2014). A synchronized visual-inertial sensor system with FPGA pre-processing for accurate real-time SLAM. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 431–437. IEEE.

[240] Nistér, D., Naroditsky, O., and Bergen, J. (2004). Visual odometry. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–652–I–659 Vol.1. IEEE.

[241] Nistér, D., Naroditsky, O., and Bergen, J. (2006). Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):3–20.

[242] Odelga, M., Stegagno, P., and Bülthoff, H. H. (2016). Obstacle Detection, Tracking and Avoidance for a Teleoperated UAV. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 2984–2990. IEEE.

[243] Olivares-Mendez, M. A., Kannan, S., and Voos, H. (2015). Vision based fuzzy control autonomous landing with UAVs: From V-REP to real experiments. In *Control and Automation (MED), 2015 23th Mediterranean Conference on*, pages 14–21. IEEE.

[244] Olivares-Méndez, M. A., Mondragón, I. F., Campoy, P., and Martínez, C. (2010). Fuzzy controller for uav-landing task using 3d-position visual estimation. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1–8. IEEE.

[245] Omari, S., Bloesch, M., Gohl, P., and Siegwart, R. (2015). Dense visual-inertial navigation system for mobile robots. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 2634–2640. IEEE.

[246] Omari, S., Gohl, P., Burri, M., Achtelik, M., and Siegwart, R. (2014). Visual industrial inspection using aerial robots. In *Applied Robotics for the Power Industry (CARPI), 2014 3rd International Conference on*, pages 1–5. IEEE.

[247] Pan, C., Hu, T., and Shen, L. (2015). BRISK based target localization for fixed-wing UAV's vision-based autonomous landing. In *Information and Automation, 2015 IEEE International Conference on*, pages 2499–2503. IEEE.

[248] Pérez-Ortiz, M., Peña, J. M., Gutiérrez, P. A., Torres-Sánchez, J., Hervás-Martínez, C., and López-Granados, F. (2016). Selecting patterns and features for between- and within- crop-row weed mapping using UAV-imagery. *Expert Systems with Applications*, 47:85–94.

[249] Pouya, S. and Saghafi, F. (2009). Autonomous Runway Alignment of Fixed-Wing Unmanned Aerial Vehicles in Landing Phase. In *Autonomic and Autonomous Systems, 2009. ICAS'09. Fifth International Conference on*, pages 208–213. IEEE.

[250] Puri, A. (2005). A Survey of Unmanned Aerial Vehicles (UAV) for Traffic Surveillance. *Department of computer science and engineering, University of South Florida*.

[251] Qingbo, G., Nan, L., and Baokui, L. (2012). The application of GPS/SINS integration based on Kalman filter. In *Control Conference (CCC), 2012 31st Chinese*, pages 4607–4610. IEEE.

[252] Qu, Y., Jiang, L., and Guo, X. (2016). Moving vehicle detection with convolutional networks in UAV videos. In *Control, Automation and Robotics (ICCAR), 2016 2nd International Conference on*, pages 225–229. IEEE.

[253] Quist, E. B. (2015). *UAV Navigation and Radar Odometry*. PhD thesis, Brigham Young University-Provo.

[254] Rathinam, S., Almeida, P., Kim, Z., Jackson, S., Tinka, A., Grossman, W., and Sengupta, R. (2007). Autonomous searching and tracking of a river using an UAV. In *American Control Conference, 2007. ACC'07*, pages 359–364. IEEE.

[255] Reardon, C. and Fink, J. (2016). Air-ground robot team surveillance of complex 3d environments. In *Safety, Security, and Rescue Robotics (SSRR), 2016 IEEE International Symposium on*, pages 320–327. IEEE.

[256] Roger-Verdeguer, J. F., Mannberg, M., and Savvaris, A. (2012). Visual odometry with failure detection for the aegis UAV. In *Imaging Systems and Techniques (IST), 2012 IEEE International Conference on*, pages 291–296. IEEE.

[257] Romero, H., Salazar, S., Santos, O., and Lozano, R. (2013). Visual odometry for autonomous outdoor flight of a quadrotor UAV. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 678–684.

[258] Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer.

[259] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE.

[260] Ruiz, J. J., Martin, J., and Ollero, A. (2015). Enhanced video camera feedback for teleoperated aerial refueling in unmanned aerial systems. In *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pages 225–231. IEEE.

[261] Saha, S., Natraj, A., and Waharte, S. (2014). A real-time monocular vision-based frontal obstacle detection and avoidance for low cost UAVs in GPS denied environment. In *Aerospace Electronics and Remote Sensing Technology (ICARES), 2014 IEEE International Conference on*, pages 189–195. IEEE.

[262] Saleem, A., Al Maashri, A., Khriji, L., and Hussein, M. (2015). An integration framework for ugv outdoor navigation system based on lidar and vision data. In *Research and Education in Mechatronics (REM), 2015 16th International Conference on*, pages 16–21. IEEE.

[263] Samadzadegan, F. and Abdi, G. (2012). Autonomous navigation of Unmanned Aerial Vehicles based on multi-sensor data fusion. In *Electrical Engineering (ICEE), 2012 20th Iranian Conference on*, pages 868–873. IEEE.

[264] Samadzadegan, F., Hahn, M., and Saeedi, S. (2007). Position estimation of aerial vehicle based on a vision aided navigation system. *Proceedings of Visualization and Exploration of Geospatial data-Stuttgart*.

[265] Sanchez-Lopez, J., Saripalli, S., Campoy, P., Pestana, J., and Fu, C. (2013). Toward visual autonomous ship board landing of a VTOL UAV. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 779–788.

[266] Sangyam, T., Laohapiengsak, P., Chongcharoen, W., and Nilkhamhang, I. (2010). Autonomous path tracking and disturbance force rejection of UAV using fuzzy based auto-tuning PID controller. In *Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010 International Conference on*, pages 528–531. IEEE.

[267] Santos, M. C., Carelli, R., and others (2015). Indoor waypoint UAV navigation using a RGB-D system. In *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pages 84–91. IEEE.

[268] Saska, M., Krajník, T., and Pfeucil, L. (2012). Cooperative μUAV-UGV autonomous indoor surveillance. In *Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on*, pages 1–6. IEEE.

[269] Scaramuzza, D. and Fraundorfer, F. (2011). Visual Odometry [Tutorial]. *IEEE robotics & automation magazine*, 18(4):80–92.

[270] Scaramuzza, D., Fraundorfer, F., and Siegwart, R. (2009). Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 4293–4299. IEEE.

[271] Scaramuzza, D. and Siegwart, R. (2008). Appearance-Guided Monocular Omnidirectional Visual Odometry for Outdoor Ground Vehicles. *IEEE transactions on robotics*, 24(5):1015–1026.

[272] Schaeffer, C. (2013). *A comparison of keypoint descriptors in the context of pedestrian detection: freak vs. surf vs. brisk*. Citeseer.

[273] Schmidt, A., Kraft, M., Fularz, M., and Domagala, Z. (2013). The comparison of point feature detectors and descriptors in the context of robot navigation. *Journal of Automation, Mobile Robotics & Intelligent Systems*, 7(1).

[274] Semsch, E., Jakob, M., Pavlicek, D., and Pechoucek, M. (2009). Autonomous uav surveillance in complex urban environments. In *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on*, volume 2, pages 82–85. IEEE.

[275] Shang, B., Liu, J., Zhao, T., and Chen, Y. (2016). Fractional order robust visual servoing control of a quadrotor UAV with larger sampling period. In *Unmanned Aircraft Systems (ICUAS), 2016 International Conference on*, pages 1228–1234. IEEE.

[276] Shang, E., An, X., Li, J., and He, H. (2014). A novel setup method of 3d LIDAR for negative obstacle detection in field environment. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 1436–1441. IEEE.

[277] Shen, S., Michael, N., and Kumar, V. (2012). Autonomous indoor 3d exploration with a micro-aerial vehicle. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 9–15. IEEE.

[278] Shi, J. et al. (1994). Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE.

[279] Shim, D. H., Chung, H., and Sastry, S. S. (2006). Conflict-free navigation in unknown urban environments. *IEEE robotics & automation magazine*, 13(3):27–33.

[280] Shirai, N. and Yamaguchi, M. K. (2004). Asymmetry in the perception of motion-in-depth. *Vision research*, 44(10):1003–1011.

[281] Shoemake, K. (1994). Euler angle conversion. In Heckbert, P. S., editor, *Graphics gems IV*, pages 222–229. Academic Press Professional, Inc.

[282] Siam, M. and ElHelw, M. (2012). Robust autonomous visual detection and tracking of moving targets in UAV imagery. In *Signal Processing (ICSP), 2012 IEEE 11th International Conference on*, volume 2, pages 1060–1066. IEEE.

[283] Soltani, H., Taghirad, H. D., and Ravari, A. N. (2012). Stereo-based visual navigation of mobile robots in unknown environments. In *Electrical Engineering (ICEE), 2012 20th Iranian Conference on*, pages 946–951. IEEE.

[284] Stoer, J. and Bulirsch, R. (2013). *Introduction to numerical analysis*, volume 12. Springer Science & Business Media.

[285] Stokkeland, M. (2014). A computer vision approach for autonomous wind turbine inspection using a multicopter.

[286] Stokkeland, M., Klausen, K., and Johansen, T. A. (2015). Autonomous visual navigation of unmanned aerial vehicle for wind turbine inspection. In *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, pages 998–1007. IEEE.

[287] Strydom, R., Thurrowgood, S., and Srinivasan, M. V. (2014). Visual odometry: autonomous uav navigation using optic flow and stereo. In *Proceedings of Australasian Conference on Robotics and Automation*.

[288] Su, Z., Wang, H., Shao, X., and Yao, P. (2015). Autonomous aerial refueling precise docking based on active disturbance rejection control. In *Industrial Electronics Society, IECON 2015-41st Annual Conference of the IEEE*, pages 004574–004578. IEEE.

[289] Sujit, P. B., Sousa, J., and Pereira, F. L. (2009a). Coordination strategies between UAV and AUVs for ocean exploration. In *Control Conference (ECC), 2009 European*, pages 115–120. IEEE.

[290] Sujit, P. B., Sousa, J., and Pereira, F. L. (2009b). UAV and AUVs coordination for ocean exploration. In *Oceans 2009-Europe*, pages 1–7. IEEE.

[291] Suzuki, S., Ishii, T., Okada, N., Iizuka, K., and Kawamura, T. (2013). Autonomous Navigation, Guidance and Control of Small Electric Helicopter. *International Journal of Advanced Robotic Systems*, 10(1):54.

[292] Tampubolon, W. and Reinhardt, W. (2014). UAV Data Processing for Large Scale Topographical Mapping. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS)*, 40(5):565–572.

[293] Tao, T. and Jia, R. (2012). UAV decision-making for maritime rescue based on Bayesian Network. In *Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on*, pages 2068–2071. IEEE.

[294] Tao, Z. and Lei, W. (2008). SINS and GPS Integrated Navigation System of a Small Unmanned Aerial Vehicle. In *Future BioMedical Information Engineering, 2008. FBIE'08. International Seminar on*, pages 465–468. IEEE.

[295] Taraldsen, G., Reinen, T. A., and Berg, T. (2011). The underwater GPS problem. In *OCEANS, 2011 IEEE-Spain*, pages 1–8. IEEE.

[296] Tardif, J.-P., Pavlidis, Y., and Daniilidis, K. (2008). Monocular visual odometry in urban environments using an omnidirectional camera. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2531–2538. IEEE.

[297] Thurrowgood, S., Soccol, D., Moore, R. J., Bland, D., and Srinivasan, M. V. (2009). A vision based system for attitude estimation of UAVs. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 5725–5730. IEEE.

[298] Todorovic, S., Nechyba, M. C., and Ifju, P. G. (2003). Sky/ground modeling for autonomous MAV flight. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 1, pages 1422–1427. IEEE.

[299] Tokekar, P., Hook, J. V., Mulla, D., and Isler, V. (2016). Sensor Planning for a Symbiotic UAV and UGV System for Precision Agriculture. *IEEE Transactions on Robotics*, pages 1–1.

[300] Tuytelaars, T., Mikolajczyk, K., et al. (2008). Local Invariant Feature Detectors: A Survey. *Foundations and trends® in computer graphics and vision*, 3(3):177–280.

[301] Valenti, R. G., Dryanovski, I., Jaramillo, C., Perea Strom, D., and Xiao, J. (2014). Autonomous quadrotor flight using onboard RGB-D visual odometry. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5233–5238. IEEE.

[302] van Blyenburgh, P. (2006). Uav systems : Global review. In *Conference, Amsterdam, The Netherlands*.

[303] Vetrella, A. R. and Fasano, B. (2016). Cooperative UAV navigation under nominal GPS coverage and in GPS-challenging environments. In *Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2016 IEEE 2nd International Forum on*, 1-5. IEEE.

[304] Vetrella, A. R., Savvaris, A., Fasano, G., and Accardo, D. (2015). RGB-D camera-based quadrotor navigation in GPS-denied and low light environments using known 3d markers. In *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, pages 185–192. IEEE.

[305] Vincenzo Angelino, C., Baraniello, V. R., and Cicala, L. (2013). High altitude UAV navigation using IMU, GPS and camera. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 647–654. IEEE.

[306] Wagtendonk, W. J. (1996). *Principles of helicopter flight*. Aviation Supplies & Academics.

[307] Waharte, S. and Trigoni, N. (2010). Supporting Search and Rescue Operations with UAVs. In *Emerging Security Technologies (EST), 2010 International Conference on*, pages 142–147. IEEE.

[308] Wang, C., Wang, T., Liang, J., Chen, Y., and Wu, Y. (2012a). Monocular vision and IMU based navigation for a small unmanned helicopter. In *2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 1694–1699.

[309] Wang, C., Zhao, H., Davoine, F., and Zha, H. (2012b). A system of automated training sample generation for visual-based car detection. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4169–4176. IEEE.

[310] Wang, G. and Ge, S. S. (2011). Robust GPS and radar sensor fusion for multiple aerial vehicles localization. In *Cybernetics and Intelligent Systems (CIS), 2011 IEEE 5th International Conference on*, pages 196–201. IEEE.

[311] Wang, Y. (2011). An Efficient Algorithm for UAV Indoor Pose Estimation Using Vanishing Geometry. In *MVA*, pages 361–364. Citeseer.

[312] Ward, S., Hensler, J., Alsalam, B., and Gonzalez, L. F. (2016). Autonomous uavs wildlife detection using thermal imaging, predictive navigation and computer vision. In *Aerospace Conference, 2016 IEEE*, pages 1–8. IEEE.

[313] Warren, M. and Upcroft, B. (2013). Robust scale initialization for long-range stereo visual odometry. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 2115–2121. IEEE.

[314] Warren, M. D. (2015). *Long-range stereo visual odometry for unmanned aerial vehicles*. PhD thesis, Queensland University of Technology.

[315] Wei, L., Cappelle, C., and Ruichek, Y. (2011). Localization of intelligent ground vehicles in outdoor urban environments using stereovision and GPS integration. In *Advanced Robotics (ICAR), 2011 15th International Conference on*, pages 192–197. IEEE.

[316] Weiss, S., Scaramuzza, D., and Siegwart, R. (2011). Monocular-SLAM-based navigation for autonomous micro helicopters in GPS-denied environments. *Journal of Field Robotics*, 28(6):854–874.

[317] Wenzel, K. E., Masselli, A., and Zell, A. (2011). Automatic Take Off, Tracking and Landing of a Miniature UAV on a Moving Carrier Vehicle. *Journal of intelligent & robotic systems*, 61(1):221–238.

[318] Williamson, W. R., Glenn, G. J., Dang, V. T., Speyer, J. L., Stecko, S. M., and Takacs, J. M. (2009). Sensor fusion applied to autonomous aerial refueling. *Journal of Guidance, Control, and Dynamics*, 32(1):262–275.

[319] Willis, A. R. and Brink, K. M. (2016). Real-time RGBD odometry for fused-state navigation systems. In *2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pages 544–552. IEEE.

[320] Wolfe, V., Frobe, W., Shrinivasan, V., and Hsieh, T.-Y. (2015). Detecting and locating cell phone signals from avalanche victims using unmanned aerial vehicles. In *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, pages 704–713. IEEE.

[321] Wu, J., Yan, J., Wang, Z., and Qu, Y. (2016). Object tracking algorithm for UAV autonomous Aerial Refueling. In *Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2016 IEEE*, pages 1665–1669. IEEE.

[322] Wu, S., Zhang, L., Xu, W., Zhou, T., and Luo, D. (2013). Docking control of autonomous aerial refueling for UAV based on LQR. In *Control and Automation (ICCA), 2013 10th IEEE International Conference on*, pages 1819–1823. IEEE.

[323] Xiang, W., Cao, Y., and Wang, Z. (2012). Automatic take-off and landing of a quad-rotor flying robot. In *Control and Decision Conference (CCDC), 2012 24th Chinese*, pages 1251–1255. IEEE.

[324] Xu, J., Solmaz, G., Rahmatizadeh, R., Turgut, D., and Boloni, L. (2015). Animal monitoring with unmanned aerial vehicle-aided wireless sensor networks. In *Local Computer Networks (LCN), 2015 IEEE 40th Conference on*, pages 125–132. IEEE.

[325] Xufeng, W., Xinmin, D., and Xingwei, K. (2013). Feature recognition and tracking of aircraft tanker and refueling drogue for UAV aerial refueling. In *Control and Decision Conference (CCDC), 2013 25th Chinese*, pages 2057–2062. IEEE.

[326] YAĞIMLI, M. and Varol, H. S. (2009). Mine detecting GPS-based unmanned ground vehicle. In *Recent Advances in Space Technologies, 2009. RAST'09. 4th International Conference on*, pages 303–306. IEEE.

[327] Yamasaki, T., Sakaida, H., Enomoto, K., Takano, H., and Baba, Y. (2007). Robust trajectory-tracking method for UAV guidance using proportional navigation. In *Control, Automation and Systems, 2007. ICCAS'07. International Conference on*, pages 1404–1409. IEEE.

[328] Yang, L., Qi, J., Xiao, J., and Yong, X. (2014). A literature review of UAV 3d path planning. In *Intelligent Control and Automation (WCICA), 2014 11th World Congress on*, pages 2376–2381. IEEE.

[329] Yang, S., Scherer, S. A., and Zell, A. (2013). An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle. *Journal of Intelligent & Robotic Systems*, 69(1-4):499–515.

[330] Yaohong, Q., Jizhi, W., Qichuan, T., and Jing, W. (2013). Autonomous alleviating loads of boom air refueling. In *Control Conference (CCC), 2013 32nd Chinese*, pages 2407–2410. IEEE.

[331] Yin, Y., Wang, X., Xu, D., Liu, F., Wang, Y., and Wu, W. (2016). Robust Visual Detection–Learning–Tracking Framework for Autonomous Aerial Refueling of UAVs. *IEEE Transactions on Instrumentation and Measurement*, 65(3):510–521.

[332] Ying-cheng, L., Dong-mei, Y., Xiao-bo, D., Chang-sheng, T., Guang-hui, W., and Tuan-hao, L. (2011). UAV Aerial Photography Technology in Island Topographic Mapping. In *Image and Data Fusion (ISIDF), 2011 International Symposium on*, pages 1–4. IEEE.

[333] Yol, A., Delabarre, B., Dame, A., Dartois, J.-E., and Marchand, E. (2014). Vision-based Absolute Localization for Unmanned Aerial Vehicles. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3429–3434. IEEE.

[334] Yoo, C.-S. and Ahn, I.-K. (2003). Low cost GPS/INS sensor fusion system for UAV navigation. In *Digital Avionics Systems Conference, 2003. DASC'03. The 22nd*, volume 2, pages 8–A. IEEE.

[335] Yoon, B.-J., Park, M.-W., and Kim, J.-H. (2006). UGV (Unmanned Ground Vehicle) navigation method using GPS and compass. In *SICE-ICASE, 2006. International Joint Conference*, pages 3621–3625. IEEE.

[336] Yu, C., Wang, J., Shan, J., and Xin, M. (2016). Multi-UAV UWA video surveillance system. In *Control, Automation, Robotics and Vision (ICARCV), 2016 14th International Conference on*, pages 1–6. IEEE.

[337] Yu, X. and Zhang, Y. (2015). Sense and avoid technologies with applications to unmanned aircraft systems: Review and prospects. *Progress in Aerospace Sciences*, 74:152–166.

[338] Yuan, C., Liu, Z., and Zhang, Y. (2015a). UAV-based forest fire detection and tracking using image processing techniques. In *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, pages 639–643. IEEE.

[339] Yuan, C., Zhang, Y., and Liu, Z. (2015b). A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques. *Canadian journal of forest research*, 45(7):783–792.

[340] Yuan, D., Yan, L., Qu, Y., and Zhao, C. (2015c). Design on decoupling control law of the refueling boom for aerial boom refueling. In *Information and Automation, 2015 IEEE International Conference on*, pages 1654–1657. IEEE.

[341] Yuan, D. L., Whidborne, J. F., and Xun, Y. J. (2014). A study on a multi-controller design of the drawtube for aerial boom refueling. In *Automation and Computing (ICAC), 2014 20th International Conference on*, pages 128–133. IEEE.

[342] Yun, B., Peng, K., and Chen, B. M. (2007). Enhancement of GPS signals for automatic control of a UAV helicopter system. In *Control and Automation, 2007. ICCA 2007. IEEE International Conference on*, pages 1185–1189. IEEE.

[343] Zedeh, L. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetic*, 3:28–44.

[344] Zeng, Q., Wang, Y., Liu, J., Chen, R., and Deng, X. (2014). Integrating monocular vision and laser point for indoor UAV SLAM. In *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2014*, pages 170–179. IEEE.

[345] Zhang, F., Stähle, H., Gaschler, A., Buckl, C., and Knoll, A. (2012). Single camera visual odometry based on Random Finite Set Statistics. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 559–566. IEEE.

[346] Zhang, J., Singh, S., and Kantor, G. (2014). Robust Monocular Visual Odometry for a Ground Vehicle in Undulating Terrain. In *Field and Service Robotics*, pages 311–326. Springer.

[347] Zhang, X., Fang, Y., Liang, X., and Zhang, X. (2016). Geometric adaptive dynamic visual servoing of a quadrotor UAV. In *Advanced Intelligent Mechatronics (AIM), 2016 IEEE International Conference on*, pages 312–317. IEEE.

[348] Zhang, X., Xian, B., Zhao, B., and Zhang, Y. (2015). Autonomous Flight Control of a Nano Quadrotor Helicopter in a GPS-Denied Environment Using On-Board Vision. *IEEE Transactions on Industrial Electronics*, 62(10):6392–6403.

[349] Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334.

[350] Zhao, X., Fei, Q., and Geng, Q. (2013). Vision based ground target tracking for rotor UAV. In *Control and Automation (ICCA), 2013 10th IEEE International Conference on*, pages 1907–1911. IEEE.

[351] Zhou, H., Kong, H., Wei, L., Creighton, D., and Nahavandi, S. (2015). Efficient Road Detection and Tracking for Unmanned Aerial Vehicle. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):297–309.

[352] Zhu, Q., Yan, Y., and Xing, Z. (2006). Robot path planning based on artificial potential field approach with simulated annealing. In *Sixth International Conference on Intelligent Systems Design and Applications*, volume 2, pages 622–627. IEEE.

[353] Ziyang, Z., Qiushi, H. A. O., Chen, G., and Ju, J. (2014). Information fusion distributed navigation for UAVs formation flight. In *Guidance, Navigation and Control Conference (CGNCC), 2014 IEEE Chinese*, pages 1520–1525. IEEE.

[354] Zogg, J.-M. (2009). Gps: Essentials of Satellite Navigation. *Compendium : Theorie and Principles of Satellite Navigation, Overview of GPS/GNSS Systems and Applications*, 58:176.